LLM-Based Semantic Entity Resolution for Intelligent Deduplication of Heterogeneous Business Lead Data: A Literature Review

Mrs. Gauri Salagare[1], Prof. Rashmi Kulkarni-Deshpande[2]
[1,2]*Department of Information Technology, Siddhant College of Engineering*

***Abstract*—Sales and marketing teams collect business leads from many places at once: web scraping, trade directories, social platforms, event lists, purchased databases and their own CRM. The same company or contact then appears several times under different spellings, formats and levels of completeness. Removing these duplicates is an entity resolution (ER) problem, and it is hard to solve with exact matching or hand-written rules. This paper reviews the research that bears on using large language models (LLMs) for that task. It traces ER from probabilistic record linkage through machine learning and pre-trained language models to prompt-based LLM matching, and it examines blocking, embeddings, prompting strategies, cost, hallucination and evaluation. The review finds that LLMs can match records with little or no labelled data and that their weaknesses, namely cost, inconsistency across pairs and unsupported answers, can be managed by placing them inside a staged pipeline. It also finds that published evaluations rely mostly on product and bibliographic benchmarks, with little evidence on business lead data. A staged architecture and an evaluation plan for lead deduplication are proposed.**

***Index Terms*—blocking, data integration, deduplication, entity resolution, large language models, lead data, record linkage, semantic matching.**

I. INTRODUCTION

A lead list is only useful if each real company or person appears once. In practice they rarely do. “Acme Industries Pvt. Ltd.”, “ACME Industries Private Limited” and “Acme Ind.” may sit in three different sources, one with a phone number, another with a website, a third with only a city. Phone numbers carry different country codes, job titles change between sources, and fields are missing. Sending the same prospect several messages waste effort and harms how a business is perceived, while merging two different firms by mistake loses a real opportunity.

The underlying task, deciding whether two records describe the same real-world entity, is known as entity resolution, record linkage or duplicate detection. It has been studied for more than half a century, beginning with the probabilistic theory of Fellegi and Sunter [1] and continuing through surveys of duplicate detection [2] and book-length treatments of data matching [3]. Recent work has opened a new line of inquiry. Narayan et al. recast data cleaning and integration tasks, including entity matching, as prompts to large foundation models and found that the models performed competitively without task-specific fine-tuning [4]. Peeters et al. later studied generative LLMs for matching in depth and argued that they depend less on task-specific training data and handle unseen entities more robustly than fine-tuned pre-trained language models [5].

These results are encouraging for lead data, which changes constantly and seldom comes with labelled match pairs. They do not settle the matter, though. Business leads differ from the product and citation records on which most studies are run. This review therefore asks three questions. What methods exist for ER, and how do LLMs fit among them? What do reported studies say about the accuracy, cost and reliability of LLM-based matching? What is missing for the specific case of heterogeneous business lead data? Section II defines the problem. Sections III to V review classical, neural and LLM-based methods. Section VI discusses the particular challenges of lead data and evaluation. Section VII synthesizes the gaps and proposes an architecture, and Section VIII concludes.

II. PROBLEM FORMULATION

Entity resolution is usually decomposed into stages. Records are first cleaned and standardized. Comparing every record with every other is quadratic in the data size, so a blocking step then selects a much smaller set of candidate pairs that are likely to match; a survey by Papadakis et al. organizes the many blocking and filtering techniques and the trade-off each makes between recall and the number of comparisons [6]. A matching step decides for each candidate pair whether the two records refer to the same entity. Finally, matched pairs are grouped into clusters, and each cluster is merged into a single consolidated record. Comparative studies of matching frameworks emphasize that the choice of attributes, similarity functions and combination strategy strongly shapes results, and that no single configuration suits every data set [7].

Two features of this decomposition matter for what follows. Errors in blocking cannot be repaired later, because a true match that is never proposed as a candidate is lost. And a pairwise decision is not the same as a consistent clustering: if record A matches B and B matches C, the system must still decide about A and C.

III. CLASSICAL APPROACHES

A. Rules, Similarity Functions and Probabilistic Linkage

The earliest practical systems combine standardization rules with string similarity measures. Names are lower-cased, legal suffixes stripped, and attributes compared with edit-based or token-based similarities. Fellegi and Sunter [1] gave a probabilistic basis for combining the agreement of several fields into a match, possible-match or non-match decision, and the framework still underlies many deduplication tools. Rule-based and probabilistic methods are fast and transparent. Their weakness is semantic: they cannot tell that “IBM” and “International Business Machines” are the same firm unless someone writes that rule, and they handle missing or conflicting fields poorly.

B. Supervised Machine Learning Pipelines

Supervised approaches learn a classifier from labelled pairs using engineered similarity features. The Magellan project packaged this workflow into a management system for entity matching, covering blocking, feature generation, labelling and debugging [8]. Such systems work well when a few hundred or thousand labelled pairs are available and the data resembles the training set, but they must be re-tuned when sources change.

IV. NEURAL AND PRE-TRAINED LANGUAGE MODEL APPROACHES

A. Deep Learning for Matching

Deep learning replaced hand-crafted features with learned representations. Mudgal et al. explored the design space of deep matching, comparing architectures that summarize and compare attributes, and found that deep models gave clear gains on textual and dirty data, while on structured, clean data they gave little over earlier methods [9]. DeepER learned distributed representations of tuples for matching and blocking [10]. These models still require substantial labelled data.

B. Transformer-Based Matchers

Pre-trained transformers brought a further step. Brunner and Stockinger applied transformer architectures to entity matching and reported strong results across benchmarks [11]. Ditto framed matching as sequence-pair classification with a pre-trained language model, serializing each record into text and adding domain knowledge and data augmentation to the fine-tuning [12]. Peeters and Bizer fine-tuned BERT with a dual objective that adds a multi-class entity signal to binary matching [13]. A survey of neural networks for entity matching summarizes these developments and notes that dependence on labelled data and difficulty with unseen entities remain open problems [14].

C. Embeddings and Vector Search for Blocking

Sentence-level encoders such as Sentence-BERT [15] produce vectors whose cosine similarity reflects semantic closeness, and they offer a natural basis for blocking: embed each record and retrieve its nearest neighbours. Zeakis et al. compared pre-trained embeddings for ER in an experimental analysis, showing that the choice of embedding model affects results across data sets [16]. Thirumuruganathan et al. studied deep learning for blocking and found that learned representations can improve candidate recall

in many settings [17]. At scale, approximate nearest-neighbour indexes make the retrieval step practical: graph-based structures such as HNSW [18] and GPU similarity-search libraries [19] are widely used for this purpose.

V. LLM-BASED ENTITY RESOLUTION

A. Prompting for Matching

In prompt-based matching, a pair of serialized records is given to an LLM together with an instruction such as "Do these two records refer to the same company?" Narayan et al. showed this to be feasible with a handful of demonstrations [4]. Peeters and Bizer examined ChatGPT for matching and reported competitive results with modest effort [20]. In a larger study, Peeters et al. evaluated several LLMs in zero-shot settings and with training data available, compared prompt designs and their sensitivity, and examined the selection of in-context demonstrations, the generation of matching rules, and fine-tuning. They also showed that GPT-4 could produce structured explanations of matching decisions and summarize classes of error, which may help engineers improve pipelines [5]. Chain-of-thought style prompting, which asks a model to reason step by step before answering [23], is one device used to elicit such explanations.

B. Cost-Aware and Collective Strategies

Calling an LLM for every candidate pair is expensive. Fan et al. studied batch prompting for ER, in which several questions share one prompt, and proposed a covering-based way of selecting demonstrations; they found batch prompting to be cost-effective against both fine-tuned language-model matchers and LLM matchers using manually designed prompts [21]. More general work on cost reduction, such as cascading cheaper and costlier models, also applies [26].

Wang et al. pointed out that most LLM matchers classify each pair independently and so ignore the global consistency among records. They compared three strategies: matching (is this pair a match?), comparing (which of two candidates is closer?) and selecting (choose the matching record from a list). Across eight ER data sets and ten LLMs, they reported that incorporating record interactions through the selecting strategy was superior, and they proposed a compound framework, ComEM, that combines strategies for better cost-effectiveness [22]. This is relevant to lead data, where one incoming record often needs to be compared with several existing candidates at once.

C. Limitations of LLM Matchers

Three limitations recur. First, LLMs can produce confident but unsupported statements, a general behaviour classed as hallucination [24]; in ER this appears as a plausible justification for a wrong merge. Second, the position of information within a long prompt affects accuracy: Liu et al. found that models use information at the start and end of long contexts better than information in the middle [25], which cautions against packing many candidates into one prompt without testing. Third, results can be sensitive to prompt wording, as the prompt-sensitivity analysis in [5] shows. A practical design must therefore bound the LLM's role, constrain its output format, and validate its decisions.

Table I. Comparison of Entity Resolution Method Families

Family	Representative works	Strength	Weakness
Rules and probabilistic	[1], [3]	Fast, transparent	Weak on semantic variation and missing data
Supervised ML	[7], [8]	Accurate with labels	Needs labelled pairs; re-tuning
Deep and transformer matchers	[9]-[14]	Strong on textual data	Needs labelled data; weaker on unseen entities
Embedding-based blocking	[15]-[19]	Semantic recall at scale	Model choice affects recall
Prompt-based LLM matching	[4], [5], [20] [22]	Little or no labelled data; explanations	Cost, prompt sensitivity, hallucination

VI. CHALLENGES SPECIFIC TO BUSINESS LEAD DATA AND EVALUATION

A. Characteristics of Lead Data

Lead data differs from commonly studied benchmarks in several ways. It is heterogeneous in schema: one source supplies a company name and a website, another a contact name, title and email. Many records are sparse, so a decision must rest on one or two fields. Entities are layered: a company has branches and subsidiaries, and a person may hold several roles, so

"same entity" needs a definition that the business agrees on. Values change over time through job moves, rebranding and relocation. Names may appear in more than one script or transliteration. These properties are discussed in the general ER literature [3], but the evaluations in [4], [5], [21] and [22] use benchmark data sets, and the sources reviewed here do not report evaluations on business lead records.

B. Benchmarks and Metrics

Early evaluations compared matching approaches on bibliographic and product data sets [27]. More recent benchmarks, such as Machamp [28] and WDC Products [29], broaden coverage to different data types and to challenges such as unseen entities and corner cases. Pairwise precision, recall and F1 are standard, but they can hide errors in the resulting clusters; Menestrina et al. discuss measures for evaluating ER results at the cluster level [30]. For a sales team the two error types carry different costs: a false merge destroys a distinct lead, while a missed duplicate causes repeated outreach. Evaluation should therefore report both and allow the threshold to be set by business preference.

C. Privacy and Governance

Lead records contain personal data such as names, emails and phone numbers. Sending them to an externally hosted LLM raises obligations under data protection law. In India, the Digital Personal Data Protection Act, 2023 [31] sets duties for those who process digital personal data. A deduplication pipeline should minimize what is sent to the model, mask direct identifiers where they are not needed for the decision, and keep an audit trail of merges.

VII. SYNTHESIS, RESEARCH GAPS AND PROPOSED ARCHITECTURE

A. Research Gaps

Table II. Research Gaps in LLM-Based Lead Deduplication

Gap	Evidence	Implication
Little evaluation on lead data	Benchmarks are product or bibliographic [4], [5], [27]-[29]	Build and report on a labelled lead data set
Pairwise decisions, inconsistent clusters	Independent pair classification [22]	Add cluster-level consistency checks
Cost at scale	Per-pair LLM calls are costly [21], [26]	Use LLMs only for ambiguous pairs
Unsupported justifications	Hallucination [24]	Constrain output; verify against fields
Entity definition and hierarchy	Company, branch, subsidiary not covered in reviewed benchmarks	Define match levels with the business
Privacy	Personal data in prompts [31]	Minimize and mask; log decisions

B. Proposed Architecture

The literature supports a staged design in which each component does what it is best at. The following is a proposal to be validated experimentally.

Standardization. Normalize case, punctuation, legal suffixes, phone formats and URLs, and parse addresses. This removes easy variation cheaply [3].

Hybrid blocking. Combine token-based keys with embedding-based nearest-neighbour retrieval [15], [18], [19] so that candidates are found despite spelling and wording differences; tune for high candidate recall [6], [17]. Deterministic matching. Resolve unambiguous pairs, such as an identical website domain or email, with rules or a light classifier [1], [8].

LLM adjudication. Send only ambiguous pairs, or one record plus a short candidate list, to the LLM with a structured prompt and a fixed JSON answer containing a decision, a confidence and the fields used as evidence [5], [22]. Batch questions where tests show it does not hurt accuracy [21]. Verification. Check that the fields cited as evidence actually agree in the source records, and route low-confidence or contradictory answers to a person. Clustering and consistency. Build clusters from accepted matches and examine chains that link records the model considered different [22], [30]. Merging with provenance. Create a consolidated record, keep the source of each value, and keep an audit trail [31].

Evaluation plan. Construct a labelled sample of lead pairs with business-defined match rules. Report pairwise precision, recall and F1, and a cluster-level measure [30]. Compare against a rules-only baseline, a fine-tuned transformer matcher [12] and an LLM-only baseline [5]. Measure cost per thousand records [21], [26], the share of pairs reaching the LLM, agreement across repeated runs and prompt variants [5], and the false-merge and missed-duplicate rates separately.

VIII. CONCLUSION

Entity resolution has moved from probabilistic rules, through supervised and neural matchers, to prompt-based matching with large language models. The reviewed work shows that LLMs can match records with little training data, give explanations, and generalize to unseen entities, and that cost-aware prompting and collective strategies improve their practicality. It also shows that they bring cost, inconsistency, prompt sensitivity and hallucination, which argue against using them alone. For heterogeneous business lead data, a staged pipeline of standardization, hybrid blocking, rules for clear cases, LLM adjudication of ambiguous ones, verification and consistent clustering is the most defensible design. The main shortfall in the evidence is empirical: there are few published evaluations on real lead data, and building a labelled lead data set with business-defined match rules would be a valuable contribution.

REFERENCES

[1]. Fellegi, I. P., and Sunter, A. B., "A theory for record linkage," Journal of the American Statistical Association, vol. 64, no. 328, pp. 1183–1210, Dec. 1969.

[2]. Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S., "Duplicate record detection: A survey," IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 1, pp. 1–16, Jan. 2007.

[3]. Christen, P., Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Berlin, Germany: Springer, 2012.

[4]. Narayan, A., Chami, I., Orr, L., Arora, S., and Ré, C., "Can foundation models wrangle your data?" Proceedings of the VLDB Endowment, vol. 16, no. 4, pp. 738–746, 2022.

[5]. Peeters, R., Steiner, A., and Bizer, C., "Entity matching using large language models," in Proceedings of the 28th International Conference on Extending Database Technology (EDBT), Barcelona, Spain, Mar. 2025, doi: 10.48786/edbt.2025.42.

[6]. Papadakis, G., Skoutas, D., Thanos, E., and Palpanas, T., "Blocking and filtering techniques for entity resolution: A survey," ACM Computing Surveys, vol. 53, no. 2, Art. no. 31, 2020.

[7]. Köpcke, H., and Rahm, E., "Frameworks for entity matching: A comparison," Data & Knowledge Engineering, vol. 69, no. 2, pp. 197–210, Feb. 2010.

[8]. Konda, P., et al., "Magellan: Toward building entity matching management systems," Proceedings of the VLDB Endowment, vol. 9, no. 12, pp. 1197–1208, 2016.

[9]. Mudgal, S., et al., "Deep learning for entity matching: A design space exploration," in Proceedings of the ACM SIGMOD International Conference on Management of Data, 2018, pp. 19–34.

[10]. Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., and Tang, N., "Distributed representations of tuples for entity resolution," Proceedings of the VLDB Endowment, vol. 11, no. 11, pp. 1454–1467, 2018.

[11]. Brunner, U., and Stockinger, K., "Entity matching with transformer architectures—A step forward in data integration," in Proceedings of the 23rd International Conference on Extending Database Technology (EDBT), 2020, pp. 463–473.

[12]. Li, Y., Li, J., Suhara, Y., Doan, A., and Tan, W.-C., "Deep entity matching with pre-trained language models," Proceedings of the VLDB Endowment, vol. 14, no. 1, pp. 50–60, 2020.

[13]. Peeters, R., and Bizer, C., "Dual-objective fine-tuning of BERT for entity matching," Proceedings of the VLDB Endowment, vol. 14, no. 10, pp. 1913–1926, 2021.

[14]. Barlaug, N., and Gulla, J. A., "Neural networks for entity matching: A survey," ACM Transactions on Knowledge Discovery from Data, vol. 15, no. 3, Art. no. 52, 2021.

[15]. Reimers, N., and Gurevych, I., "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3982–3992.

[16]. Zeakis, A., Papadakis, G., Skoutas, D., and Koubarakis, M., "Pre-trained embeddings for entity resolution: An experimental analysis," Proceedings of the VLDB Endowment, vol. 16, no. 9, pp. 2225–2238, 2023.

[17]. Thirumuruganathan, S., et al., "Deep learning for blocking in entity matching: A design space exploration," Proceedings of the VLDB Endowment, vol. 14, no. 11, pp. 2459–2472, 2021.

[18]. Malkov, Y. A., and Yashunin, D. A., "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 824–836, Apr. 2020.

[19]. Johnson, J., Douze, M., and Jégou, H., "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, Jul. 2021.

[20]. Peeters, R., and Bizer, C., "Using ChatGPT for entity matching," arXiv:2305.03423, 2023.

[21]. Fan, M., Han, X., Fan, J., Chai, C., Tang, N., Li, G., and Du, X., "Cost-effective in-context learning for entity resolution: A design space exploration," in Proceedings of the IEEE 40th International Conference on Data Engineering (ICDE), 2024, doi: 10.1109/ICDE60146.2024.00284.

[22]. Wang, T., Chen, X., Lin, H., Chen, X., Han, X., Sun, L., Wang, H., and Zeng, Z., "Match, compare, or select? An investigation of large language models for entity matching," in Proceedings of the 31st International Conference on Computational Linguistics (COLING), Abu Dhabi, UAE, Jan. 2025, pp. 96–109.

[23]. Wei, J., et al., "Chain-of-thought prompting elicits reasoning in large language models," in Advances in Neural Information Processing Systems (NeurIPS), vol. 35, 2022, pp. 24824–24837.

[24]. Ji, Z., et al., "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, Art. no. 248, pp. 1–38, 2023.

[25]. Liu, N. F., et al., "Lost in the middle: How language models use long contexts," Transactions of the Association for Computational Linguistics, vol. 12, pp. 157–173, 2024, doi: 10.1162/tacl_a_00638.