Group, Embed and Reason: A Hybrid LLM and Embedding Framework for Semantic Attribute Alignment

Shramona Chakraborty[*], **Shashank Mujumdar**[*], **Nitin Gupta**[*], **Sameep Mehta**, **Ronen Kat**, **Itay Etelis**, **Mohamed Mahameed**, **Itai Guez**, **Rachel Brill**
IBM Research
shramona.chakraborty1@ibm.com, shamujum@in.ibm.com, ngupta47@in.ibm.com, sameepmehta@in.ibm.com, ronenkat@il.ibm.com, itay.etelis@ibm.com, mohamed.mahameed@ibm.com, itai.guez@ibm.com, rachelt@il.ibm.com

Abstract

In enterprise systems, tasks like API integration, ETL pipeline creation, customer record merging, and data consolidation rely on accurately aligning attributes that refer to the same real-world concept but differ across schemas. This semantic attribute alignment is critical for enabling schema unification, reporting, and analytics. The challenge is amplified in schema only settings where no instance data is available due to ambiguous names, inconsistent descriptions, and varied naming conventions.

We propose a hybrid, unsupervised framework that combines the contextual reasoning of Large Language Models (LLMs) with the stability of embedding-based similarity and schema grouping to address token limitations and hallucinations. Our method operates solely on metadata and scales to large schemas by grouping attributes and refining LLM outputs through embedding-based enhancement, justification filtering, and ranking. Experiments on real-world healthcare schemas show strong performance, highlighting the effectiveness of the framework in privacy-constrained scenarios.

1 Introduction

In modern data integration workflows such as constructing ETL pipelines across heterogeneous sources, interfacing APIs between third-party systems, or merging customer records across internal business units, the successful integration depends on identifying and aligning semantically equivalent fields across schemas (Ceri et al., 2003). These attributes may differ in name, structure, or format but refer to the same underlying concept. For instance, in human resources data, `annual_salary` in one schema may correspond to `yearly_income` in another. Although lexically distinct, these attributes share the same semantic intent, and failure to correctly align them compromises downstream analytics and decision making. This foundational task, known as semantic mapping, ensures that integrated data remains interpretable and consistent across systems.

Despite its critical role, automating semantic mapping remains a long-standing challenge. Schema heterogeneity manifested as inconsistent naming conventions, varying schema design philosophies, and domain-specific terminologies introduces ambiguity. For example, `is_contractor` may encode the same information as `employment_type`, or `last_purchased_item` may align with `most_recent_transaction`. These relationships require contextual reasoning to detect and cannot be resolved by surface-level comparisons. This problem is further exacerbated in privacy-sensitive domains such as healthcare and finance, where access to instance-level data is often restricted due to regulatory and compliance constraints. As a result, semantic mappings must be inferred using only schema-level metadata, including attribute names and short natural language descriptions—which are frequently sparse, inconsistent, or under-specified.

Traditional approaches rely on string similarity metrics or heuristic rules that operate on lexical cues. These methods are fast and interpretable but perform poorly when faced with semantic equivalence that lacks lexical overlap (Rahm and Bernstein, 2001). Embedding based methods improve on this by representing attribute names and descriptions as vectors in a high dimensional semantic space, enabling comparison through cosine similarity (Cappuzzo et al., 2020). These techniques can align fields like `Temp_C` and `t_celsius`, where surface overlap is limited but semantic similarity is preserved in the embeddings. However, such methods still fall short when the required alignment depends on logical inference or deeper contextual understanding, as in aligning `is_contractor` to `employment_type`, or mapping purchasing behav-

[*]Authors contributed equally to the work

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1703–1710

ior fields across domains.

Recent advances in large language models (LLMs) offer new opportunities for tackling this problem. LLMs can perform zero-shot reasoning over natural language, allowing them to identify conceptual equivalences even when attribute names differ significantly. However, they are constrained by fixed input token limits, which makes them unsuitable for directly processing large schemas end-to-end, a common scenario in enterprise environments like ERP systems, customer data platforms, or healthcare registries. They are also prone to hallucinating mappings that appear plausible but lack grounding in the input schema, or generating additional matches based on overly loose or generic justifications. Additionally, while LLMs excel at contextual inference, they may overlook surface-level alignments that are semantically valid but underexplained, necessitating complementary mechanisms to recover or verify these matches.

To address the limitations of prior approaches in schema-only semantic mapping at scale, we propose a hybrid framework that combines the contextual reasoning capabilities of LLMs with the semantic robustness of embedding-based similarity. The framework consists of four key stages:

- Clustering and grouping of similar attributes in order to fit with LLM token limits.
- Produce candidates based on attribute names and descriptions using LLMs.
- Embedding similarity to (i) recover high-confidence mappings missed by the LLM and (ii) filter out hallucinated or semantically weak predictions.
- Post processing to remove hallucination and inconsistencies, and to rank the candidates.

This modular pipeline enables robust, unsupervised alignment of semantically equivalent attributes in schema-only settings, without relying on labeled data or instance-level values.

2 Related Work

Semantic attribute mapping has long been studied across database, data integration, and knowledge representation communities. Early approaches relied on syntactic similarity measures such as edit distance, token overlap, and naming heuristics (Rahm and Bernstein, 2001). While efficient and interpretable, these methods are limited in handling semantically equivalent but lexically divergent attributes (e.g., `is_smoker` vs. `smoking_status`).

Embedding-based techniques (Cappuzzo et al., 2020) improved upon this by leveraging vector representations of attribute names and descriptions to capture semantic similarity. Though these methods achieve higher recall than string matching, they often fail to infer deeper relationships that require contextual understanding. Supervised deep models (e.g., SMAT (Zhang et al., 2021)) use BiLSTMs with attention to model attribute pairs. While effective, they rely on labeled data, limiting generalizability across unseen schemas.

More recently, LLM-based methods have been explored for schema alignment (Sheetrit et al., 2024), demonstrating improved contextual reasoning. Some variants use retrieval-augmented generation, treating schema elements as documents and fetching relevant context from external sources. Although these methods enhance mapping quality, they often assume access to instance-level data or external corpora, introducing infrastructure complexity and limiting applicability in privacy-sensitive, schema-only settings.

To our knowledge, no prior work addresses LLM-based semantic mapping under strict schema-only conditions. We propose a fully unsupervised hybrid framework that clusters attributes using embeddings, invokes LLMs within token limits, refines predictions through similarity scoring, and prunes hallucinations via justification quality, achieving scalable and accurate alignment without training data or instance values.

3 Proposed Framework

3.1 Problem Formulation

Let $S = \{s_1, s_2, \ldots, s_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ denote the set of attributes in the *source schema* and *target schema* respectively. Each attribute $s_i \in S$ or $t_j \in T$ is associated with schema-level metadata in the form of a tuple (n_i, d_i), where n_i is the attribute name and d_i is a short textual description (if available). No instance-level data or external ontologies are assumed to be available. The goal is to identify a set of semantic mappings:

$$M \subseteq (S \cup \{\emptyset\}) \times (T \cup \{\emptyset\})$$

where each mapping $(s_i, t_j) \in M$ indicates that the source attribute s_i and target attribute t_j are semantically equivalent or closely related in meaning. The objective is to construct the mapping set M as

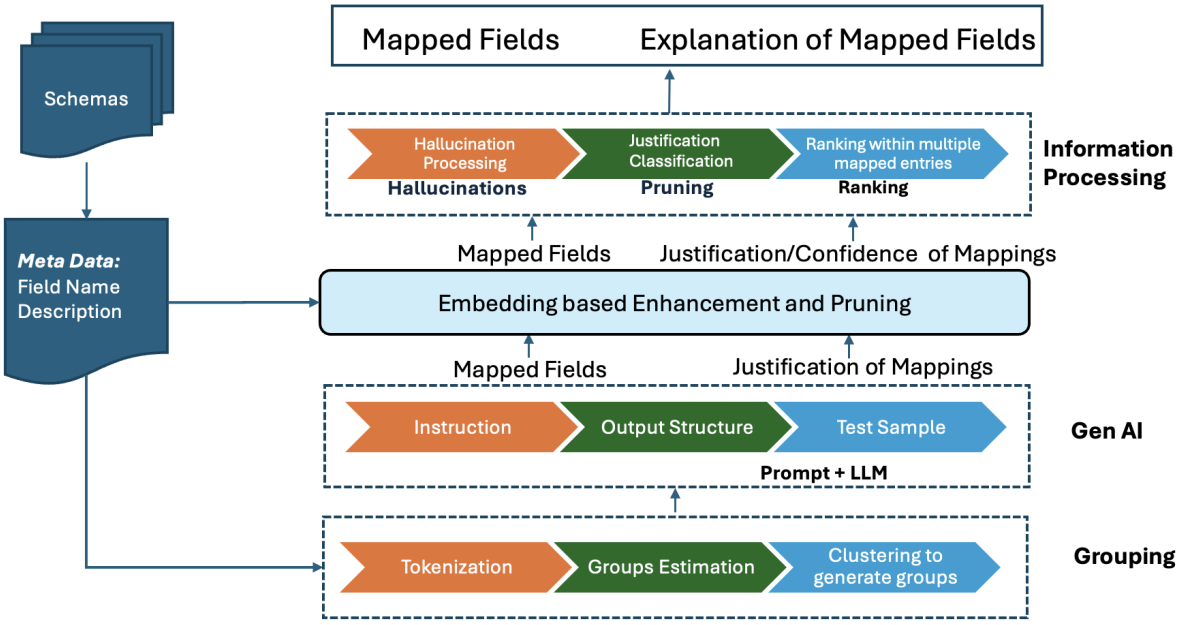

Figure 1: Proposed Framework for Semantic Mapping.

accurately and comprehensively as possible using only schema-level metadata. We explicitly allow the following types of mappings:

- **One-to-one mappings:** Semantic equivalence between a source and a target attribute.
- **One-to-many / Many-to-one mappings:** Cases where a single attribute semantically corresponds to multiple attributes in the other schema.
- **Empty mappings:** Attributes from either schema with no valid semantic counterpart. These are captured as $(s_i, \emptyset)$ or $(\emptyset, t_j)$ in source or target schema, respectively.

3.2 Framework Overview

Our proposed framework addresses the semantic mapping problem in a fully unsupervised and scalable manner using a multi-stage architecture as shown in Figure 1. It is composed of four stages:

1. **Semantic Grouping:** Partitioning schema attributes into semantically coherent subsets using embedding-based clustering to adhere to the token limit of the language model.
2. **GenAI-based Mapping Generation:** Employing a LLM to generate candidate semantic mappings between grouped source and target attributes using only metadata such as names and descriptions.
3. **Embedding-Based Enhancement and Pruning:** Enhancing and filtering the LLM-generated mappings by computing embedding-based similarity scores, thereby removing semantically weak or inconsistent pairs and adding semantically strong pairs (missed by LLM).
4. **Information Processing and Final Mapping Selection:** Post-processing the filtered mappings to remove hallucinations, prune unjustified alignments, and rank candidates.

3.2.1 Semantic Grouping

Large language models (LLMs) have limited context windows, making it impractical to align large schemas directly. To address this, we partition the source and target schemas into semantically coherent groups using embedding-based similarity.

Let $a \in S \cup T$, and let $\mathbf{e}_a \in \mathbb{R}^d$ denote the embedding of attribute a. We define a similarity matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$:

$$\mathbf{M}_{ij} = \frac{\mathbf{e}_{s_i} \cdot \mathbf{e}_{t_j}}{\|\mathbf{e}_{s_i}\| \, \|\mathbf{e}_{t_j}\|}$$

We apply clustering (k-Nearest-Neighbors) on the resulting matrix to generate k groups $\{G_1, \ldots, G_k\}$. The number of groups is determined by the LLM token limit and average token length per attribute. This ensures each group fits within the LLM's input window while preserving contextual coherence.

3.2.2 GenAI-Based Mapping Generation

Once grouping is complete, each cluster of source and target attributes is provided to a LLM through carefully designed prompts as shown below. The

LLM is instructed to generate candidate semantic mappings using only the available attribute names and descriptions. Prompts are optimized to elicit high-quality outputs and may include structural patterns or few-shot demonstrations to encourage consistency and reduce ambiguity. The LLM produces a set of predicted mappings along with natural language justifications, which are retained for downstream validation.

Formally, let $G_k = (S_k, T_k)$ denote the k-th group of source and target attribute subsets after grouping. For each group, we define the LLM as a function:

$$\text{LLM}(G_k) \rightarrow \{((s_i, t_j), r_{ij}) \mid s_i \in S_k,\ t_j \in T_k\}$$

where each output consists of a candidate attribute pair (s_i, t_j) and its corresponding justification r_{ij} in natural language. This structured output supports downstream modules such as plausibility filtering and explanation-based pruning.

Mapping Prompt

Instruction: You are a knowledgeable assistant skilled at matching fields between different data sources. Your task is to ensure that the fields are mapped correctly and include explanations for each mapping choice. Use the sourceName.fieldName format to specify each source field clearly.

Few important points to consider while mapping -
(a) Avoid mixing fields from different sourceNames.
(b) Do not map within same schema.
(c) In case of non related mapping, return None.
(d) In case of multiple mappings return all valid mappings.

Sample Format -

```
{
"schemas": [
{
"sourceName": <sourceName1>,
"properties": { ... }
},
{
"sourceName": <sourceName2>,
"properties": { ... }
},
{
"sourceName": "<sourceName3>",
"properties": { ... }
}
]
}
```

Response: { ... }

Test Example -
Input: { ... }

Response:

3.2.3 Embedding-Based Enhancement and Pruning

In parallel to the LLM-based mapping generation, we compute semantic similarity scores between all source and target attribute pairs using static embedding models (all-mpnet-base-v2 (Sentence-Transformers, 2021a)). These scores are used in two complementary roles:

- **Enhancement:** For each pair $(s_i, t_j) \in S \times T$, we compute the cosine similarity $(sim(s_i, t_j))$. If $\text{sim}(s_i, t_j) \geq \theta_{\text{add}}$ (0.8) and $(s_i, t_j) \notin M_{\text{LLM}}$, the pair is added to the candidate mapping set:

$$M_{\text{enhanced}} \leftarrow M_{\text{LLM}} \cup \{(s_i, t_j)\}$$

  Along with the mapping, a simple justification is attached based on the embedding score: The justification is generated as: "s_i *and* t_j *are related with a confidence of sim*(s_i, t_j)".
  This enhancement allows recovery of semantically correct mappings that may have been omitted by the LLM due to input limitations or ambiguous prompt interpretation.
- **Pruning:** If $(s_i, t_j) \in M_{\text{LLM}}$ but $\text{sim}(s_i, t_j) < \theta_{\text{drop}}(0.2)$, the mapping is removed:

$$M_{\text{pruned}} \leftarrow M_{\text{enhanced}} \setminus \{(s_i, t_j)\}$$

  This helps eliminate hallucinated or spurious alignments where no meaningful semantic relationship exists.

The thresholds θ_{add} and θ_{drop} are globally applied and selected through empirical tuning. By combining LLM inference with embedding-based validation and justification, this hybrid step enhances both the quality and interpretability of the semantic mapping output.

3.2.4 Information Processing

The final stage focuses on refining and validating the remaining set of candidate mappings. This stage includes three steps:

- **Hallucination Removal:** We identify and eliminate mappings where the LLM introduces attribute pairs not present in the actual source or target schemas. We define the hallucination-free mapping set as:

$$M_{\text{filtered}} = \{(s_i, t_j) \in M_{\text{pruned}} \mid s_i \in S \wedge t_j \in T\}$$

  Any pair $(s_i, t_j) \in M_{\text{pruned}}$ where $s_i \notin S$ or $t_j \notin T$ is considered a hallucination and is removed from the final output.
- **Justification Pruning:** Each LLM-generated mapping $(s_i, t_j) \in M_{\text{filtered}}$ is associated with a natural language justification $J_{i,j}$. To assess its plausibility, we apply a zero-shot entailment classifier f_{ZSC} with candidate labels: "`strongly related`", "`maybe related`", and

Dataset	String Similarity				Embedding-Based			Proposed (LLM-Based)						
	JW	LV	ME	NG	MPNet-dot	MPNet-base	DistilRoBERTa	Granite	LLaMA-3	Mistral	DeepSeek	LLaMA-4	Phi	Qwen
CMS	0.149	0.396	0.392	0.397	0.422	0.391	0.428	0.528	0.561	0.590	**0.638**	0.513	0.547	0.584
SAKI	0.339	0.411	0.389	0.405	0.396	0.383	0.375	0.677	0.727	0.746	0.686	**0.792**	0.558	0.735
Synthea	0.221	0.352	0.341	0.339	0.344	0.355	0.411	0.644	0.674	0.703	0.732	0.661	**0.745**	0.684
MIMIC-III	0.244	0.383	0.386	0.384	0.349	0.345	0.388	0.564	0.645	0.614	**0.713**	0.628	0.659	0.649
MIMIC-IV	0.218	0.415	0.415	0.404	0.409	0.411	0.441	0.506	0.510	0.589	0.600	0.542	**0.611**	0.553

Table 1: Performance (F1) comparison across full datasets and methods. Jaro-Wiker - JW, Levenshtein - LV, Monge-Elkan - ME, N-Gram - NG.

"`unrelated`". The classifier returns a confidence score for each label, and we retain a mapping only if the confidence score for "`strongly related`" exceeds a threshold θ_{entail}:

$$\text{score}_{i,j} = f_{\text{ZSC}}(J_{i,j}; \text{“strongly related”})$$

$$M_{\text{just}} \mathrel{+}= \{(s_i, t_j) \in M_{\text{filtered}} \mid \text{score}_{i,j} \geq \theta_{\text{entail}}\}$$

The resulting set M_{just} is then passed to the ranking module.

- **Ranking:** In cases where multiple candidate mappings exist for a given source attribute, we apply an embedding-based ranking mechanism. For each candidate target, we compute the cosine similarity. This process scores and orders candidates based on their semantic similarity in the embedding space, prioritizing mappings that are more closely aligned with the source attribute meaning.

These information processing steps help ensure the reliability and precision of the final output. By combining structured filtering and ranking mechanisms with LLM reasoning, the system produces high-quality semantic mappings in a fully unsupervised setting. The overall design remains modular and adaptable, allowing for flexibility across domains, schema sizes, and environments.

4 Experiments and Results

We evaluate on five schema-only datasets from real and synthetic healthcare sources, varying in size, complexity, and metadata quality (Table 2). Each dataset is a source–target schema pair with attribute names and descriptions; no instance-level data is used due to privacy constraints. **CMS:** (Zhang et al., 2021) Large-scale Medicare schema with minimal metadata, posing a low-context challenge. **SAKI:** (Zhang et al., 2021) Curated interoperability schemas with moderate metadata and semantic ambiguities. **MIMIC-III:** (Sheetrit et al., 2024) ICU dataset with structured schema and rich metadata. **MIMIC-IV:** (Parciak et al., 2024) Updated MIMIC-III with restructured schema and new fields. **Synthea:** (Zhang et al., 2021) Synthetic EHR with clean metadata, enabling evaluation under ideal and complex mapping scenarios.

CMS	SAKI	MIMIC-III	MIMIC-IV	Synthea
38	24	26	9	12

Table 2: No. of schema pairs across different datasets.

We use expert-aligned mappings where available; all pairs include at least one verifiable match.

4.1 Evaluation Metrics

We evaluate alignment performance using the *F1-score*. Each source attribute is treated as a classification instance with a predicted mapping (s, t), where s is either a source attribute or the empty token ϵ (no mapping).

We define binary classification outcomes as:

- **True Positive (TP):** A non-empty predicted mapping (s, t) correctly matches the ground truth.
- **False Positive (FP):** A non-empty predicted mapping (s, t) does not exist in the ground truth.
- **False Negative (FN):** A ground truth mapping exists, but either no prediction or no correct prediction is made.
- **True Negative (TN):** Both prediction and ground truth specify no mapping (i.e., $t = \emptyset$).

We compute F1-scores for both the *matched* and *unmatched* classes, and report the final result as their macro-average. This strategy accounts for alignment correctness while penalizing spurious predictions which is crucial in schema-only settings where many attributes lack valid mappings.

4.2 Baselines

To evaluate the effectiveness of our framework, we compare against representative baselines spanning embedding-based, and classical string similarity methods:

- **Embedding-Based Models**: We compute cosine similarity between attribute embeddings using pre-trained sentence models—*all-mpnet-*

Dataset	Target Table	Target Column	Target Description	Source Table	Source Column	Source Description	GT	Pred	Comments
CMS	beneficiarysummary	bene_death_dt	date of death	death	death_datetime	the date and time the person was deceased.	0	1	Erroneous GT
CMS	carrierclaims	line_alowd_chrg_amt	line allowed charge amount 1	visit_occurrence	provider_id	a foreign key to the provider in the provider table who was associated with the visit.	1	0	Erroneous GT
SYNTHEA	patients	address	patient's street address without commas or newlines.	person	location_id	a foreign key to the place of residency for the person in the location table, where the detailed address information is stored.	0	1	This kind of Indirect mapping may benefit the user.
CMS	inpatientclaims	clm_pmt_amt	claim payment amount	procedure_occurrence	quantity	the quantity of procedures ordered or administered.	0	0	LLM generated it; Embedding Pruning removed it.
CMS	carrierclaims	clm_thru_dt	claims end date	death	death_datetime	the date and time the person was deceased.	0	0	LLM generated it; Justification pruning removed it.
CMS	carrierclaims	clm_from_dt	claims start date	provider	-	-	-	-	Mapped within the same schema instead of the provider schema column. LLM hallucinated; Hallucination Removal module fixed it.

Table 3: Representative examples of alignment predictions showing ground truth inconsistencies and the filtering effect of embedding and justification modules.

base-v2 (Sentence-Transformers, 2021a), *multi-qa-mpnet-base-dot-v1* (Sentence-Transformers, 2021b), and *all-distilroberta-v1* (Sentence-Transformers, 2022)—and apply greedy alignment. These models capture contextual similarity but may miss implicit or inferential mappings.

- **String Similarity Methods** (Rahm and Bernstein, 2001): Classical string-based techniques (Jaro-Winkler, Levenshtein, Monge-Elkan, N-gram) are used to score attribute pairs.

4.3 Results and Discussions

4.3.1 Comparison with String Similarity and Embedding Based Approaches

We evaluate all approaches including string similarity methods, embedding-based baselines, and our proposed framework on a full dataset. For the string similarity and embedding-based approaches, we perform a grid search over similarity thresholds ranging from 0.0 to 1.0 in increments of 0.1. For each method, the threshold yielding the best performance across the datasets is used for reporting. Our proposed framework is evaluated using multiple language models, including Granite (granite-3-8b-instruct (IBM Granite, 2024)), LLaMA3 (Llama-3.3-70B (Meta, n.d)), Mistral (mistral-large (Mistral AI, 2024)), DeepSeek (DeepSeek-V3 (DeepSeek-AI et al., 2025)), LLaMA4 (llama-4-maverick-17b (Meta)), Phi (phi-4 (Microsoft)), and Qwen (qwen2.5-72B-instruct (Qwen)) to assess model-agnostic performance and robustness across LLM variants.

Results are shown in Table 1, where our framework demonstrates consistent improvements ranging from 7% to 40% across datasets and model variants. Overall, the DeepSeek family outperforms other models, likely due to its stronger instruction-following capabilities, reduced hallucination behavior, and superior performance on reasoning tasks. This makes it particularly well-suited for structured alignment tasks where concise, high-quality generation is essential.

4.3.2 Comparison with Supervised Approach

To contextualize the effectiveness of our unsupervised approach, we also evaluated it against SMAT, a supervised model trained on labeled mappings. While this is not a strictly fair comparison—since SMAT benefits from supervised training and task-specific tuning—it provides a useful benchmark. On the same dataset splits, our method achieved F1-scores within 12–18% of SMAT, despite operating in a zero-shot setting using only schema metadata. We believe that this performance gap could be further narrowed, or even surpassed, with stronger LLMs such as GPT-4, which offer improved semantic reasoning and language understanding. We scoped our evaluation to lower-cost, open-weight, computationally feasible models.

4.3.3 Impact of Embedding Filtering and Justification Pruning on Performance

Across all evaluated datasets and language models, we observed a clear and consistent pattern regarding the contributions of Embedding Filtering (EF) and Justification Pruning (JP). The complete framework, incorporating both EF and JP, consistently achieves the highest performance scores. When either EF or JP is individually removed, performance declines moderately by approximately 2–5%, indicating that each component independently enhances the model's effectiveness. The simultaneous removal of both EF and JP results in the

largest degradation, with performance drops ranging from about 7–10% relative to the full configuration. These findings collectively affirm that both EF and JP play complementary and essential roles in achieving robust and optimal performance across diverse datasets and model architectures.

4.3.4 Qualitative Analysis of Results

Beyond quantitative metrics, we manually inspected selected predictions to assess ground truth (GT) issues, and module contributions.

(1) Ground Truth (GT) Issues - We observed several GT inconsistencies that affected F1 scores:

- The model produced semantically correct mappings absent from the GT (e.g., `bene_death_dt` → `death_datetime`; row 1), marked as false positives.
- Some GT alignments were weak or incorrect (e.g., `line_alowd_chrg_amt` → `provider_id`).
- In cases with missing GT (e.g., row 3), the model generated reasonable mappings like `address` → `location_id`, inferred via implicit semantics.

(2) Effect of Modules - The proposed layered architecture helped suppress weak predictions:

- Row 4: Spurious mapping (`clm_pmt_amt` → `procedure_occurrence`) was filtered by *Embedding Pruning*.
- Row 5: Weak temporal link (`clm_thru_dt` → `death_datetime`) was discarded by *Justification Pruning*.
- Row 6: LLM hallucinated a same-schema mapping (`clm_from_dt` → `clm_thru_dt`) instead of cross-schema; corrected by *Hallucination Removal*.

These cases demonstrate the importance of layered filtering in ensuring robustness. While the LLM captures high-recall candidates, embedding and justification pruning help eliminate weak or hallucinated predictions, contributing to overall precision improvements.

5 Conclusion

We presented a fully unsupervised framework for schema-only semantic mapping that combines the interpretability of embedding-based similarity with the contextual reasoning capabilities of large language models (LLMs). By introducing an adaptive grouping strategy that respects token constraints, and a filtering pipeline to mitigate hallucinations, our method provides a scalable and accurate solution for aligning schema attributes without requiring training data or access to instance-level values.

Evaluations on real-world schemas show that this hybrid approach effectively handles diverse mapping challenges without requiring training data or instance values. Our filtering module, prunes implausible LLM outputs and ranks candidates based on semantic plausibility, further improves the reliability and precision of mappings. Our method enables accurate, unsupervised schema alignment in a privacy-sensitive setting, offering a practical solution for real-world data integration tasks.

Limitations

While our framework is effective in handling schema-only mapping tasks, it has a few limitations. First, the reliance on textual metadata (names and descriptions) makes it sensitive to poorly documented or inconsistently labeled schemas. When descriptions are too sparse or ambiguous, neither embeddings nor LLMs may resolve the correct semantic match without additional context.

Second, although our grouping strategy helps overcome token limits, grouping quality directly affects LLM output. Semantically incoherent groups may lead to suboptimal or incomplete mappings. Future work could explore more sophisticated group selection mechanisms that incorporate ontology-based or retrieval-augmented context.

Third, while our hallucination and justification pruning modules improve mapping reliability, they may also remove valid mappings when the LLM's explanation lacks sufficient clarity or semantic coherence. This can occur in cases where the domain-specific synonyms, abbreviations, or implicit relationships are triggered, where the correct mapping is generated but the accompanying justification is weak or generic. As our filtering logic relies partly on the quality of these explanations, improving the robustness of justification scoring or incorporating fallback reasoning from structured knowledge could help preserve such mappings without compromising precision.

Finally, although our approach does not require labeled training data, it also does not learn from user feedback or mapping outcomes over time. Incorporating feedback loops or lightweight fine-tuning to adapt the system across domains remains an open direction for enhancing long-term robustness and generalizability.

References

Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2020. Creating embeddings of heterogeneous relational datasets for data integration tasks. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data*.

Stefano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, and Maristella Matera. 2003. *Morgan Kaufmann series in data management systems: Designing data-intensive Web applications*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report.

IBM Granite. 2024. Granite 3.0 8b instruct. https://huggingface.co/ibm-granite/granite-3.0-8b-instruct.

Llama 3 Meta. n.d. Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/.

Llama 4 Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

Phi 4 Microsoft. Introducing phi-4: Microsoft's newest small language model specializing in complex reasoning. https://shorturl.at/BoMf0.

Mistral AI. 2024. Mistral large instruct. https://huggingface.co/mistralai/Mistral-Large-Instruct-2407.

Marcel Parciak, Brecht Vandevoort, Frank Neven, Liesbet M Peeters, and Stijn Vansummeren. 2024. Schema matching with large language models: an experimental study. *arXiv preprint arXiv:2407.11852*.

2.5 Qwen. Qwen2.5: A party of foundation models! https://qwenlm.github.io/blog/qwen2.5/.

Erhard Rahm and Philip A Bernstein. 2001. A survey of approaches to automatic schema matching. *the VLDB Journal*.

Sentence-Transformers. 2021a. All mpnet base v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

Sentence-Transformers. 2021b. Multi-qa-mpnet-base-dot-v1. https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1.

Sentence-Transformers. 2022. All distilroberta v1. https://huggingface.co/sentence-transformers/all-distilroberta-v1.

Eitam Sheetrit, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2024. Rematch: Retrieval enhanced schema matching with llms. *arXiv preprint arXiv:2403.01567*.

Jing Zhang, Bonggun Shin, Jinho D Choi, and Joyce C Ho. 2021. Smat: An attention-based deep learning solution to the automation of schema matching. In *Advances in Databases and Information Systems: 25th European Conference, ADBIS 2021, Tartu, Estonia, August 24–26, 2021, Proceedings 25*.