

MONTAGE: LLM-driven Expert-in-loop Schema-Ontology Mapper

Siddhesh Thombre¹, Manasi Patwardhan¹ and Sunita Sarawagi²

¹TCS Research

²IIT Bombay

Abstract

Integrating heterogeneous relational databases into a centralized ontology remains a persistent challenge in enterprise knowledge representation, primarily due to discrepancies in schema structures, semantic interpretations, entity naming conventions, and incomplete/noisy schema or ontology. Performing database-to-ontology mapping manually is slow, error-prone and cognitively taxing at scale. Applying LLMs through direct prompting to generate complete end-to-end mappings often yields poor results. We introduce MONTAGE, an LLM-driven, human-in-the-loop mapper that adopts a neuro-symbolic design to decompose the mapping problem into cascaded sub-tasks and leverage self-generated demonstrations guided by domain-agnostic patterns to supervise each sub-task. The system blends automation with guided user feedback to refine and validate outputs, thereby enhancing mapping accuracy, reducing cognitive effort, and substantially accelerating the integration. The demo lets users upload schema - ontology pair, review and refine intermediate outputs such as ontology-database alignments and SQL views, and finally generate and validate the complete R2RML mappings.

Keywords

Database-Ontology mapping, data integration, enterprise search, R2RML generation

1. Introduction

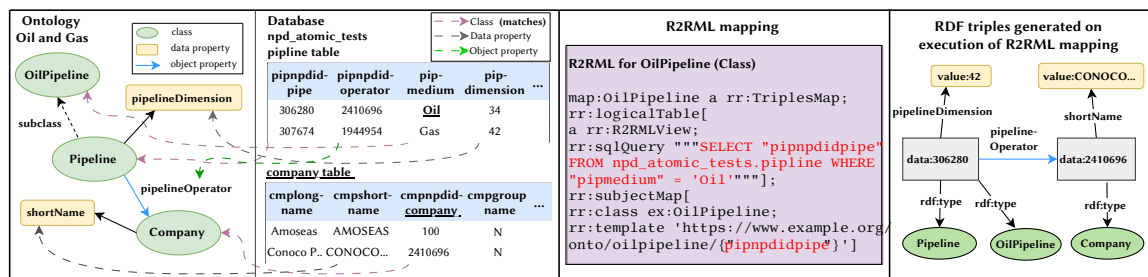


Figure 1: R2RML Generation (DB-to-Ontology Mapping) Example

Enterprises seek to represent organizational knowledge using unified ontologies to ensure semantic consistency across systems [1]. However, enterprise data is typically distributed across heterogeneous relational databases. Mapping these databases to a centralized ontology is essential for effective querying but is time-consuming, error-prone, and cognitively demanding at enterprise scale, where ontologies may include hundreds of entities and databases hundreds of tables and thousands of columns. Traditional automated approaches and tools [2, 3, 4, 5] rely largely on surface-level string matching, which fails to capture deeper semantic relationships, domain-specific correspondences, and cryptically named schema elements. Large Language Models (LLMs) offer stronger semantic understanding, but direct prompting to generate complete database-to-ontology mappings yield poor results. Full automation is rarely feasible, as enterprise schemas embed implicit enterprise-specific domain knowledge that is not explicitly stated yet crucial for correct alignment. Expert oversight is therefore indispensable, making a expert-in-the-loop paradigm the right fit.

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

✉ siddhesh.thombre@tcs.com (S. Thombre); manasi.patwardhan@tcs.com (M. Patwardhan); sunita@iitb.ac.in (S. Sarawagi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

We introduce MONTAGE, a database-to-ontology mapping tool that combines LLM-driven automated mapping generation with expert guidance. MONTAGE follows an interactive, feedback-driven workflow that reduces effort while improving mapping accuracy and cost-effectiveness. Mappings are expressed in R2RML, where logical tables defined over SQL views enable the transformation of relational data into RDF triples in the target ontology (Figure 1 illustrates an example). Since database-ontology mapping is a one-time, infrequent activity whose outputs are reused across many downstream queries, the upfront expert-in-the-loop effort is justified; we therefore prioritize accuracy over compute cost to ensure high-quality mappings.

The key components of MONTAGE are as follows:

Neuro-symbolic task decomposition: A two-stage pipeline consisting of *Matching* and *Mapping*. Ontology concepts are aligned with database tables and columns using a top-down process with symbolic pruning, followed by SQL view and R2RML generation. LLMs are invoked at scoped sub-stages, and intermediate results are presented for expert verification to reduce cognitive load and error propagation.

Pattern-based self-demonstrations: To improve LLM performance via in-context learning, MONTAGE automatically generates few-shot exemplars using pre-defined, domain-agnostic matching and mapping patterns, eliminating the need for manual demonstration curation.

Pattern induction from expert feedback: MONTAGE learns novel patterns from user-corrected instances and adds them to the pattern repository, increasing the diversity of auto-generated demonstrations and improving sub-task performance.

Confidence-based adaptive LLM invocations: For each ontology concept, candidate database entities are ranked with confidence scores aggregated from multiple LLM calls. The number of LLM invocations is adaptively adjusted, fewer for high-confidence cases and more for uncertain ones, to balance robustness and cost.

2. Overview of MONTAGE

The design of MONTAGE is motivated by the observation that current LLMs, despite their long-context capabilities, struggle to accurately generate R2RML mappings through direct prompting (see Table 2). To address this limitation, MONTAGE decomposes the task into two sequential sub-tasks: matching and mapping. Each sub-task is performed using LLMs with stage-specific in-context examples and contextualization to capture inter-stage dependencies. Figure 2 presents the overall workflow of MONTAGE, which begins with users loading the ontology and database files. Intermediate results are presented at every stage for validation, enabling users to review and edit outputs before proceeding, thereby preventing error propagation. The matching and mapping stages are executed in batches, with results displayed after each batch to gather user feedback. MONTAGE incorporates this feedback by converting expert edits into reusable signals, improving accuracy and reducing user cognitive load in subsequent batches. The following sections elaborate on the key components of MONTAGE.

2.1. Neuro-Symbolic Decomposition

Data Enrichment: Optionally, MONTAGE infers missing foreign keys in the database and missing domain or range declarations in the ontology before matching and mapping stages, improving downstream alignment quality.

Matching: The matching stage aligns ontology concepts with database entities using LLM-based ranking over symbolically pruned candidate sets. It proceeds sequentially through class, data property, and object property matching, guided by predefined rules that ensure coverage while constraining the search space such as the following: An ontology class is matched to one or more tables via primary key columns, optionally filtered by categorical values, which define candidates for subsequent stages. Data properties are matched to columns in class-associated or referencing tables, while object properties are matched either to relevant columns in domain or range tables or to primary key pairs in junction tables. Each sub-stage is executed batch-wise over subsets of concepts, and final matches are selected based on a confidence threshold (Section 2.4).

Mapping: In the mapping stage, matched concepts are converted into SQL views through class, data property, and object property mapping. Outputs are generated batch-wise and include SQL views with subject and object column mappings, which are presented to users for validation and refinement. The refined views are then used to produce R2RML mappings that generate RDF triples to populate the ontology. MONTAGE supports R2RML mapping verification through natural language queries. LLMs translate these queries into SQL and SPARQL, enabling users to compare results over the relational database and the generated RDF data to validate the mappings.

2.2. Pattern based Self-Demonstrations

Zero-shot LLM calls for schema matching and mapping underperformed (see "-Self-Demonstrations(FS)" in Table 2), motivating a shift to in-context learning (ICL) with few-shot demonstrations. Manually curated examples substantially improved accuracy (Table 2) but are labor-intensive and not scalable due to the need for domain and task expertise. Automatic bootstrapping method utilized in Matchmaker [6, 7] yielded inferior results (Table 2) because the generated demonstrations were semantically homogeneous, limiting coverage of real scenarios. To combine scalability with quality, we introduce a pattern-driven generator for any DB-ontology pair.

At the outset of the matching and mapping, the user is presented with a set of predefined patterns, which encapsulate a typical matching (Table 1) or mapping behavior and is agnostic to the ontology domain. The users can select relevant patterns or add missing patterns, after which MONTAGE automatically generates database and ontology specific demonstrations as instances of the chosen patterns, which serve few shots, conditioning on outputs from prior stages to improve quality and coverage (e.g., data-property and object-property matching few-shots are synthesized using class-matching results as input). This design prioritizes diversity, ensuring robust coverage across varied cases.

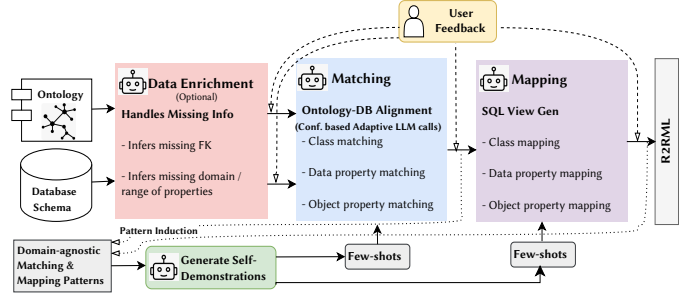


Figure 2: Overview of the MONTAGE workflow with three stages, Data Enrichment, Matching, and Mapping, incorporating user feedback after each stage.

2.3. Pattern Induction from Expert Feedback

MONTAGE automatically learns novel patterns from user corrected matches, and adds them to the pattern repository so that subsequent batches include more diverse examples tied to these newly discovered patterns, improving accuracy over batches.

2.4. Confidence based Adaptive LLM Invocations

During the matching stage of MONTAGE, each sub-stage produces a ranked list of candidate matches with associated confidence scores. Confidence is estimated by aggregating results from multiple LLM invocations that vary in prompt phrasing and example selection. We find this consensus-based approach to be more reliable than relying on confidence signals from a single invocation. Importantly, these confidence scores are meaningful indicators rather than arbitrary values: we consistently observe that low-confidence predictions often correspond to incorrect mappings. For instance, for the MainArea ontology class, the top-ranked candidate license received a low confidence score of 41, while the correct match was a different entity. To limit overhead, MONTAGE adapts the number of LLM invocations based on confidence levels. A single invocation is used when the top-ranked candidate has high confidence, while additional invocations are triggered incrementally, up to a maximum of four, when confidence is low. The confidence threshold is adjusted batch-wise based on prior outcomes. It is increased when user edits are frequent and decreased when many concepts yield no database matches.

Table 1
Matching Patterns for Self Demonstration

Pattern name	Pattern definition
<i>Class matching</i>	
No Match	No corresponding table found for a class.
Exact or Syntactic match	Table name identical or syntactically derived from the class name.
Semantic match	Same meaning between table and class name.
Domain match	Matching requires domain knowledge.
Categorical column match	Class represents a subset of table data identified by a specific categorical value.
<i>Data Property matching</i>	
Column of the matched table	Match with a column of the domain class’s table.
Column of the related table	Match with a column of a table referencing the domain table.
<i>Object Property matching</i>	
Foreign key of domain table	Match with a foreign key in the domain table referencing the range table.
Foreign key of range table	Match with a foreign key in the range table referencing the domain table.
Junction table	Match with columns of a junction table linking the domain and range tables.

Table 2

Results on RODI Benchmark with Gemini-2.0-Flash.

Conf: Conference-no-fk; NPD: Npd-Atomic;

FS: Few-Shot; #: With programmatic mapping;

†: No Decomposition

Scenario	Conf	Mondial	NPD	Avg
Traditional Baselines				
AM4O [3]	0.41	0.44	0.23	0.26
Milan [2]	0.46	–	0.30	0.31
LLM Driven Baselines				
LLM4VKG [8]	0.51	0.18	0.19	0.21
Table-to-KG [9]#	0.33	0.10	0.17	0.18
Magneto [10]#	0.41	0.28	0.21	0.23
LLM Driven Demonstration Based Baselines				
Matchmaker [6, 7]	0.56	0.38	0.19	0.24
Manually Curated FS	0.72	0.82	0.48	0.53
Our Approach	0.79	0.79	0.51	0.56
Ablations				
- Self-Demonstrations (FS)	0.54	0.46	0.36	0.38
- Matching	0.51	0.50	0.36	0.38
- Mapping#	0.62	0.36	0.43	0.44
- Matching+Mapping†	0.46	0.08	0.07	0.10

3. Results

We evaluate MONTAGE on three challenging RODI benchmark scenarios from the conference, geographical, and oil-and-gas domains using the standard local F-measure over semantically equivalent SPARQL–SQL test query pairs. As shown in Table 2, MONTAGE achieves the best overall performance with F1 scores of 0.79 on Conference-no-fk, 0.79 on Mondial, and 0.51 on NPD-Atomic, yielding the highest average score of 0.56. The reported results are based on the top-ranked matches above the confidence threshold, using a single invocation per concept. It consistently outperforms traditional systems and prior LLM-based methods, showing that neither schema-driven techniques nor direct LLM prompting alone suffice for accurate end-to-end mapping. MONTAGE’s strong performance stems from its staged decomposition into matching and mapping, which reduces search space, improves contextualization, and integrates symbolic constraints with LLM reasoning. Ablation results confirm this design choice, as removing stages or collapsing them into a single step causes substantial drops in performance, with one-shot mapping performing worst. Pattern-guided self-demonstration drives significant performance gains while eliminating the need for manually curated few-shot examples. Expert evaluation shows that MONTAGE reduces manual effort by approximately 45–70% while delivering at-par or better mapping quality.

4. Conclusion

We presented MONTAGE, an LLM-driven, expert-in-the-loop system for database-to-ontology mapping that combines neuro-symbolic decomposition with pattern-guided self-demonstrations, along with guided user feedback to improve mapping quality. Through the NPD scenario (with approximately 70 tables and 1,000 columns mapped to an ontology of about 650 entities), the demo demonstrates how MONTAGE supports users across Matching and Mapping stages via an interactive browser-based interface. The system enables experts to review intermediate outputs, inspect ranked alternatives with confidence scores and reasoning, and refine generated SQL views and R2RML mappings. An

accompanying demo video and prompts are available at the provided link¹.

The demonstration highlights how structuring the workflow into staged, reviewable steps focuses user attention on low-confidence cases, thereby reducing cognitive load and improving mapping accuracy. It also supports validation through natural language queries posed over both the source database and the mapped RDF data.

As future work, we aim to reduce cost by complementing LLM-based reasoning with programmatic and template-driven strategies for well-structured and repetitive patterns, while retaining LLM flexibility for complex cases.

Declaration on Generative AI

During the preparation of this work, Microsoft Copilot was used to paraphrase and reword text. The author(s) reviewed and edited the content and take full responsibility for the final publication.

References

- [1] M. Buron, F. Goasdoué, I. Manolescu, M.-L. Mugnier, Obi-wan: ontology-based rdf integration of heterogeneous data, *Proc. VLDB Endow.* 13 (2020) 2933–2936. URL: <https://doi.org/10.14778/3415478.3415512>. doi:10.14778/3415478.3415512.
- [2] S. Mathur, D. O’Sullivan, R. Brennan, Milan: Automatic generation of r2rml mappings, in: *Irish Conference on Artificial Intelligence and Cognitive Science*, 2018. URL: <https://api.semanticscholar.org/CorpusID:49570848>.
- [3] Á. Sicilia, G. Nemirovski, Automap4obda: Automated generation of r2rml mappings for obda, in: *International Conference Knowledge Engineering and Knowledge Management*, 2016. URL: <https://api.semanticscholar.org/CorpusID:38363184>.
- [4] J. Bak, M. Blinkiewicz, Square: A visual tool for creating r2rml mappings, in: *International Workshop on the Semantic Web*, 2016. URL: <https://api.semanticscholar.org/CorpusID:17632067>.
- [5] Á. Sicilia, G. Nemirovski, A. Nolle, Map-on: A web-based editor for visual ontology mapping, *Semantic Web* 8 (2017) 969–980.
- [6] N. Seedat, M. van der Schaar, Matchmaker: Self-improving large language model programs for schema matching, *ArXiv abs/2410.24105* (2024). URL: <https://api.semanticscholar.org/CorpusID:273707861>.
- [7] N. Seedat, M. van der Schaar, Bootstrapping self-improvement of language model programs for zero-shot schema matching, in: *Forty-second International Conference on Machine Learning*, 2025.
- [8] G. Xiao, L. Ren, G. Qi, H. Xue, M. D. Panfilio, D. Lanti, Llm4vkg: Leveraging large language models for virtual knowledge graph construction, in: *International Joint Conference on Artificial Intelligence*, 2025. URL: <https://api.semanticscholar.org/CorpusID:281477725>.
- [9] N. Vandemoortele, B. Steenwinckel, S. V. Hoecke, F. Ongenae, Scalable table-to-knowledge graph matching from metadata using llms, in: *SemTab@ISWC*, 2024. URL: <https://api.semanticscholar.org/CorpusID:275571188>.
- [10] Y. Liu, E. H. M. Pena, A. S. R. Santos, E. Wu, J. Freire, Magneto: Combining small and large language models for schema matching, *ArXiv abs/2412.08194* (2024). URL: <https://api.semanticscholar.org/CorpusID:274638317>.

¹<https://drive.google.com/drive/folders/1mh2vIS4QwonMnOh3xvLd3Q1SV8Cffgkm>