Aligning Real-World Document Schemas to Fine-Grained Concept Vocabularies

Enver Menadjiev
KAIST
Daejeon, Republic of Korea
enver@kaist.ac.kr

Isaac Fisher
L2 Labs
New York, NY, USA
isaac@l2labs.ai

Kai Spicer
L2 Labs
Columbia University
New York, NY, USA
kes2246@columbia.edu

Christos Koutras
New York University
New York, NY, USA
christos.koutras@nyu.edu

Andrew Bell
L2 Labs
Cornell Tech
New York, NY, USA
andrew@l2labs.ai

Abstract

Schema matching, or aligning semantically related elements across schemas, is a foundational task in data integration and interoperability. In this short paper, we report on a year-long effort to align attributes in XML documents to a fine-grained concept vocabulary. This problem is especially important in insurance, where organizations exchange policy and claims data through legacy, proprietary, and product-specific XML schemas that must be reconciled for interoperability. This setting is understudied and differs from existing schema matching benchmarks: source schemas contain local conventions and custom business logic, while the target vocabulary has been built up over decades by many contributors, leading to inconsistencies. Building on recent work that combines smaller fine-tuned language models with larger frontier models, we propose an agent-based architecture for schema matching. This paper discusses the industrial constraints that shaped the system, failure modes of retrieval-and-re-ranking systems, and practical lessons for schema matching in the wild.

Keywords

Schema Matching, Data Integration, XML, Canonical Data Models, Insurance, LLM Agents

1 Introduction

Aligning schemas remains a major challenge in enterprise data systems, including business-to-business (B2B) integrations, interoperability, and database migration. Classical schema matching systems combine lexical, structural, and instance-level signals to identify correspondences between schemas [5–7], and modern benchmarks have made these methods easier to compare [3]. More recently, LLMs and AI agents have renewed interest in automating schema matching and related data integration tasks [2, 8, 9], yet current methods still lack granular evaluation in real-world applications [2].

Several recent LLM-based schema matching approaches decompose the problem into candidate retrieval followed by LLM-based selection or re-ranking. Most related to our work is Liu et al. [4], who use fine-tuned Small Language Models (SLMs)—such as models from the BERT family [1]—for *candidate retrieval*, followed by a Large Language Model (LLM) for *re-ranking*. We initially deployed this approach but found that it decreased performance relative to our baseline, highlighting the need to test schema matching methods in realistic enterprise settings.

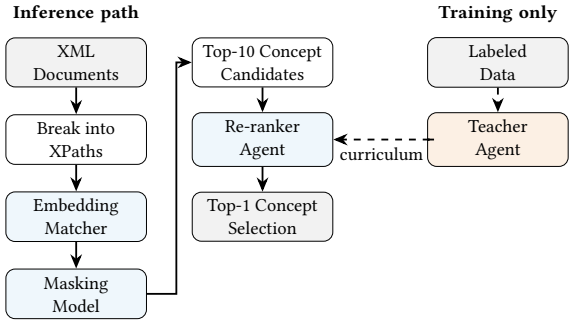

Figure 1: Pipeline for matching XPaths to concepts using ML for retrieval and a teacher/re-ranker agent for selection.

2 Setting

We study a setting encountered in a production deployment with a large enterprise company in which an integration engineer must match attributes from XML documents[1] to a canonical *concept vocabulary*: a list of fields representing semantic business ideas.

We explore this problem in insurance using ACORD, an organization that defines common data formats for insurance information exchange used by over 36,000 organizations worldwide. Although ACORD standardizes insurance data, organizations still vary in their schemas, naming conventions, and product-specific extensions. This makes insurance a natural stress test: the same construct can appear across workflows with different names, nesting structures, and interpretations.

Since we cannot release proprietary business data, we synthesize 49 XML documents reflecting field observations. The task is to align these heterogeneous schemas to a shared, fine-grained concept vocabulary containing over 2,000 possible semantic fields.

[1] XML documents represent data as a tree of nested attributes, forming a schema.

Formalization. Let $\mathcal{X}$ be the space of XML documents, C the concept vocabulary, and $\mathcal{M}$ the space of match sets. We define schema matching as $\mathcal{X} \times C \to \mathcal{M}$, where each $M \in \mathcal{M}$ consists of tuples (x_i, c_j): x_i is an XPath[2] and $c_j \in C$ is a concept. Each XPath is matched to at most one concept. For example:

$$\text{AcordDocument/PolicyBind/Classification/Code/Value} \to \text{RISK\_SUBJECT\_NAICS\_CD} \quad (1)$$

Challenges. This setting is deceptively simple. We identify three sources of difficulty: (1) some correct matches have little semantic overlap and depend entirely on custom business logic; (2) the vocabulary contains "false friends," where the semantic and business meanings are different; and (3) the fine-grained vocabulary has been built over decades by multiple engineers, leaving some concepts poorly delineated.

3 Methodology

Following Liu et al. [4], we decompose schema matching into candidate retrieval and re-ranking; Figure 1 shows the full architecture.[3] Given an XML document, our system extracts XPaths x_i and maps each to one concept $c_j \in C$. Supervised XPath–concept training captures recurring mappings, while an agentic re-ranker uses context and historical usage to resolve ambiguous cases.

Our synthetic dataset assigns each XPath a ground-truth concept and hand-labeled masks over XPath segments: $x_{i,j}$ denotes the j-th '/'-delimited segment of XPath x_i, and $m_{i,j} \in \{0, 1\}$ indicates whether that segment is important for identifying the correct concept.

Deterministic Candidate Generation. We train a matcher p that maps XPaths and concepts into a shared embedding space using an encoder f trained with Multiple Negatives Ranking Loss [10]. At inference, we score each XPath–concept pair by cosine similarity,

$$p(x_i, c_j) = \text{sim}(f(x_i), f(c_j)), \quad (2)$$

and retrieve the top-k concepts for each XPath. We then train a supervised masking model g to identify informative XPath segments and construct a masked representation $s(x_i)$. The final deterministic score combines full- and masked-XPath evidence:

$$z(x_i, c_j) = \alpha p(x_i, c_j) + (1 - \alpha) p(s(x_i), c_j), \quad (3)$$

where $\alpha \in [0, 1]$ weights the full-XPath score. Sorting candidates by z gives the top-k list passed to the agent.

Agentic Teaching and Re-Ranking. We found that at $k = 10$, the deterministic stage often retrieves the correct concept but fails to rank it first, motivating an agentic re-ranker over a small candidate set. An agent is well suited to re-ranking because it can make complex, non-linear decisions resembling human review. The re-ranker agent has three tools: one for inspecting document-level structure, one for comparing candidates against prior labeled examples, and one for querying aggregate concept usage patterns before selecting a final match. We further introduce a teacher-curriculum paradigm: during training, a teacher evaluates the re-ranker's decisions and corrects mistakes on labeled training documents, then converts feedback into a curriculum that is provided as context during validation inference. This loop gives the agent reusable context to solve for specific conventions, false friends, and custom business logic.

[2] An XPath is a path expression that identifies a node in an XML document tree.

[3] An open-source implementation of our approach and the synthetic data can be found at https://github.com/L2-AI/schema-concept-alignment

Table 1: Accuracy over all XML documents. ML top-10 is the recoverable ceiling; Δ columns are relative to ML top-1.

ML top-1	ML top-10	Magneto Δ	Agent Δ
64.59%	93.19%	−8.22 pp	**+4.78 pp**

4 Experiments

We evaluate on the 49-document synthetic corpus using leave-one-out validation: for each held-out document, the deterministic model is trained on the remaining documents, and the agent receives curriculum feedback only from labeled training documents.

Table 1 reports top-1 accuracy for the ML baseline, a Magneto-style single-call LLM re-ranker, and our agentic re-ranker; ML top-10 is the recoverable ceiling. The ML model retrieves the correct concept in its top-10 candidates 93.19% of the time, but ranks it first only 64.59% of the time. Magneto decreases top-1 accuracy by 8.22 points relative to the ML baseline, while our agent improves it by 4.78 points, suggesting that structured context and curriculum feedback are more effective than one-shot re-ranking. Overall we find that enterprise schema matching is not just a retrieval problem; when candidate recall is high but top-1 accuracy is low, the bottleneck becomes context-aware decision-making.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[2] Juliana Freire, Grace Fan, Benjamin Feuer, Christos Koutras, Yurong Liu, Eduardo Peña, Aécio Santos, Cláudio T Silva, and Eden Wu. 2025. Large language models for data discovery and integration: Challenges and opportunities. *IEEE Data Engineering Bulletin* (2025).

[3] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. 2021. Valentine: Evaluating matching techniques for dataset discovery. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 468–479.

[4] Yurong Liu, Eduardo Pena, Aecio Santos, Eden Wu, and Juliana Freire. 2024. Magneto: Combining small and large language models for schema matching. *arXiv preprint arXiv:2412.08194* (2024).

[5] Jayant Madhavan, Philip A Bernstein, Erhard Rahm, et al. 2001. Generic schema matching with cupid. In *vldb*, Vol. 1. 49–58.

[6] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. 2002. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings 18th international conference on data engineering*. IEEE, 117–128.

[7] Erhard Rahm and Philip A Bernstein. 2001. A survey of approaches to automatic schema matching. *the VLDB Journal* 10, 4 (2001), 334–350.

[8] Aécio Santos, Eduardo HM Pena, Roque Lopez, and Juliana Freire. 2025. Interactive Data Harmonization with LLM Agents: Opportunities and Challenges. *arXiv preprint arXiv:2502.07132* (2025).

[9] Aaron Steiner and Christian Bizer. 2026. Automatic End-to-End Data Integration using Large Language Models. *arXiv preprint arXiv:2603.10547* (2026).

[10] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018). arXiv:1807.03748 http://arxiv.org/abs/1807.03748