Exploiting Mapping Corpora and Expert Knowledge for LLM-based Schema Mapping Generation

Yannis Marketakis[1,2,*,†], Angely Buckel[3,†] and Yannis Tzitzikas[1,2,†]

[1]Institute of Computer Science, FORTH-ICS Heraklion, Crete, Greece
[2]Computer Science Department, University of Crete Heraklion, Crete, Greece
[3]École nationale d'ingénieurs de Brest, Brest, France

Abstract

Schema mappings are essential for transforming heterogeneous data into knowledge graphs, but their manual construction requires significant expertise in the source structure, the target ontology, and the mapping language. While large language models (LLMs) can generate mapping drafts, weakly guided generation often leads to incomplete, syntactically invalid, or semantically inaccurate mappings. This paper investigates how different forms of guidance can support LLM-based schema mapping generation, focusing on the X3ML mapping language, while proposing a methodology that can also be applied to other declarative mapping languages. The examined guidance mechanisms cover complementary aspects of the mapping task, including the syntactic structure of the mapping language, explanatory documentation of the target ontology, reusable mapping examples extracted from existing X3ML projects and explicit modelling rules provided by experts. The approaches are evaluated experimentally against gold standard mappings from X3MLBench using quantitative metrics, complemented by an analysis of common failure modes. The results provide insights into how syntactic, ontological, retrieval-based, and expert-provided guidance affect the accuracy, completeness, and reliability of LLM-generated schema mappings.

Keywords

schema mapping, X3ML mapping language, LLM-based mapping generation, retrieval-augmented generation, expert-guided schema mappings

1. Introduction

The construction of knowledge graphs from heterogeneous data sources typically requires the definition of schema mappings that specify how source data structures are transformed into instances of a target ontology. These mappings are a critical part of the overall transformation process, as they determine not only the syntactic conversion of data into RDF, but also the semantic interpretation of source elements according to the classes and properties of the target ontology. In practice, schema mapping construction remains a demanding and time-consuming task. It requires good understanding of the source schema, familiarity with the modelling principles of the target ontology, and knowledge of the mapping language used to express the transformation.

Recent advances in Large Language Models (LLMs) [1] have created new opportunities for supporting knowledge graph construction. Several approaches explore the direct use of LLMs for transforming heterogeneous inputs into RDF representations. However, such prompt-to-RDF approaches may limit transparency, reproducibility, and maintainability, since the modelling decisions that lead to the produced RDF graph remain implicit and difficult to validate or reuse. For this reason, recent work [2] has argued that LLMs should be repositioned from black-box graph transformers to assistive generators of explicit schema mappings, while the actual data transformation should remain delegated to established transformation frameworks.

Graph-enhanced LLMs for trustwOrthy Web data management (GLOW), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy

*Corresponding author.
†These authors contributed equally.
marketak@ics.forth.gr (Y. Marketakis); a25buckel@enib.fr (A. Buckel); tzitzik@ics.forth.gr (Y. Tzitzikas)
https://users.ics.forth.gr/~marketak (Y. Marketakis); https://users.ics.forth.gr/~tzitzik/ (Y. Tzitzikas)
0000-0002-0417-2526 (Y. Marketakis); 0000-0001-8847-2130 (Y. Tzitzikas)



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

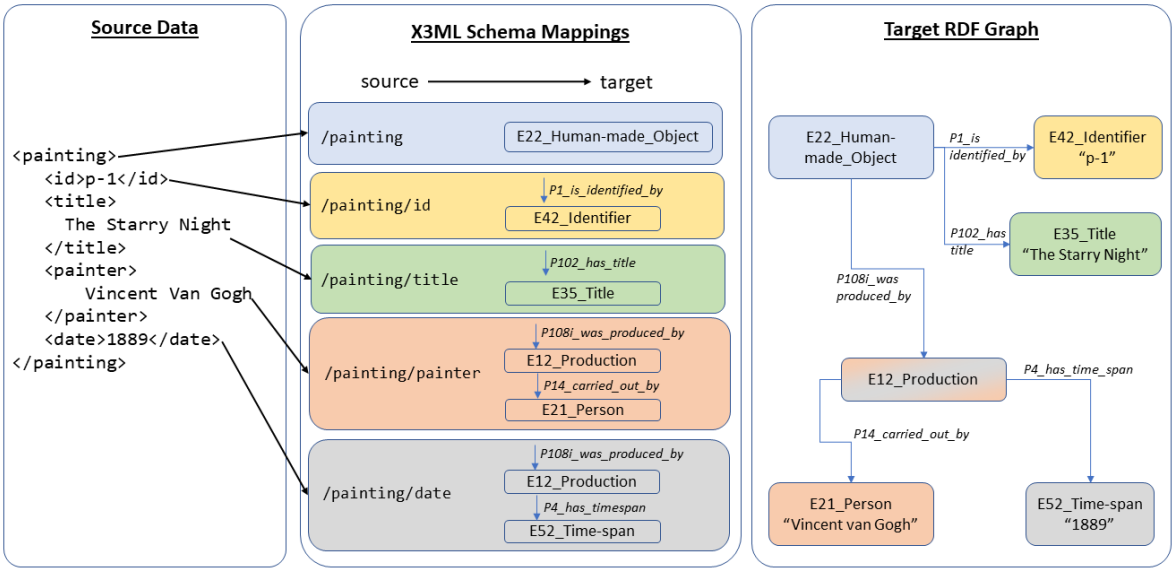


Figure 1: Simplified example of an X3ML schema mapping, illustrating how fields from an XML record are mapped to classes and properties of the target ontology, here CIDOC CRM, producing an RDF graph.

X3ML [3] is a declarative mapping language for transforming structured data into RDF knowledge graphs. It defines mappings between source schemas, such as XML structures, and target ontologies, while also supporting URI generation policies and controlled transformation workflows. However, high-quality X3ML mappings still depend heavily on human expertise, since mapping designers must identify relevant source elements, select appropriate ontology classes and properties, and ensure that the produced mappings are syntactically valid and semantically meaningful.

Figure 1 illustrates the basic role of an X3ML mapping through a simplified and diagrammatic example. The left part of the figure shows a structured source record describing a painting, including fields such as its identifier, title, painter, and date. The middle part shows the corresponding X3ML mapping rules, where source paths (demonstrated as XPath expressions) are associated with constructs (demonstrated as RDF class-property paths) of the target ontology, which in this example is the ISO 21127:2014 CIDOC CRM [4]. These mappings do not rename source fields; they define target ontology paths that specify how source values should be represented as instances of CIDOC CRM classes and connected through particular properties. For example, the `painting` element is mapped to `E22_Human-made_Object`, while the `title` element is represented as `E35_Title`, and connected through `P102_has_title`. The right part of the figure shows the resulting RDF graph produced by applying these mappings. For ease of understanding, the X3ML mappings in Figure 1 are shown diagrammatically; in practice, they are serialized into XML and can be created manually or with the support of the 3M Editor[1]. Concrete examples are available through the X3ML GitHub repository[2].

The feasibility of using LLMs for schema mapping generation has been investigated in prior work [5], where the task was framed as a constrained structured generation problem involving syntactic, semantic, and ontological constraints. That work introduced X3MLBench, a benchmark of real-world, expert-validated X3ML mapping projects, and evaluated different LLMs and prompting strategies for generating mappings in the context of ontology-based data integration. The results showed that LLMs can generate syntactically valid and partially accurate mappings, but also revealed recurring limitations related to ontology alignment, structural consistency, and the handling of ambiguous or weakly structured inputs. That work showed that LLM performance depends critically on the availability of structured examples and domain-relevant context.

Building on these findings, in this paper we focus on the role of guidance in LLM-based schema

[1] https://demos.isl.ics.forth.gr/3m/
[2] https://github.com/isl/x3ml/

mapping generation. The main hypothesis explored in this work is that guided schema mapping generation can produce more accurate and practically useful results than unguided or weakly guided generation. In this context, guidance refers to additional information that helps the LLM understand the mapping language, the target ontology, relevant previous mappings, or expert modelling decisions. Such guidance can restrict the space of possible outputs, expose the model to reusable mapping patterns, and improve its ability to select appropriate classes and properties. Therefore, the paper does not merely examine whether LLMs can generate X3ML mappings, but investigates how different forms of guidance affect the quality of the generated mappings.

To investigate the role of guidance in LLM-based schema mapping generation, we examine five alternative approaches: a syntax-only baseline, an ontology-informed approach, a RAG-based approach exploiting previous X3ML mappings, a correspondence-enhanced RAG approach that combines semantic retrieval with lexical re-ranking, and an expert-rule-based approach. These approaches allow us to compare how syntactic, ontological, retrieval-based, and expert-provided guidance affect the quality of generated X3ML mappings.

Although the experiments in this paper focus on X3ML, the proposed methodology is not limited to a single mapping language. The same general principle can be applied to other declarative mapping languages, provided that suitable syntactic specifications, ontology documentation, example mappings or expert rules are available. In this sense, X3ML is used as a concrete and realistic case study for investigating a broader question: *how can LLMs be effectively guided to generate explicit, reusable, and semantically meaningful schema mappings for knowledge graph construction?*

The paper evaluates the five approaches through a set of experimental mapping scenarios using X3MLBench [5]. The generated mappings are compared against reference mappings using quantitative evaluation measures that capture their correspondence with the expected output. This allows us to assess whether the different forms of guidance lead to measurable improvements over the syntax-only baseline. In addition to the quantitative evaluation, we analyze recurring failure modes in the generated mappings, including missing correspondences, unsupported mappings, incorrect ontology classes and properties, invalid X3ML structures, and hallucinated ontology terms. Overall, the main contribution of this paper is an experimental study of guidance strategies for LLM-based schema mapping generation. More specifically, the paper: (i) defines and compares five approaches with different levels of guidance for generating X3ML mappings; (ii) investigates the use of ontology documentation, retrieved mapping examples, and expert rules as sources of semantic and modelling support; (iii) evaluates the generated mappings through automatic comparison with reference mappings; and (iv) discusses the strengths, limitations, and practical implications of each approach for semi-automatic schema mapping construction.

2. Background and Related Work

Schema mapping generation is a central problem in data integration and knowledge graph construction. Declarative mapping languages provide an explicit and reusable way to specify how heterogeneous source data are transformed into RDF graphs conforming to a target ontology. Representative examples include R2RML [6] for relational databases, RML [7] for heterogeneous structured sources such as CSV, JSON, and XML, YARRRML [8] as a more human-readable syntax for RML/R2RML mappings, and X3ML [3]. Despite the availability of such languages and tools, high-quality mappings remain largely manual and expertise-intensive, requiring knowledge of the source structure, the target ontology, and the mapping language. More broadly, recent surveys on LLM-empowered knowledge graph construction [9] show that LLMs are increasingly used across ontology engineering and graph construction tasks.

Recent work has explored the use of LLMs for supporting schema mapping generation. Schmidt et al. [10] investigate LLM-supported and context-enhanced generation of RML and YARRRML mappings for manufacturing knowledge graphs, evaluating the generated mappings against expert-authored references. Hofer et al. [11] study the automatic generation of RML mappings for semi-structured data using a small ontology derived from DBPedia resources. Karkee et al. [12] combine RML and LLMs from

a different perspective, by embedding LLM calls within RML mappings, in order to extract ontological terms from semi-structured or textual data during transformation.

The use of LLMs in relation to X3ML has been investigated in our previous work. A first vision paper [2] argued against relying on LLMs as direct prompt-to-RDF transformers and proposed repositioning them as assistants for generating explicit schema mappings. A subsequent study [5] introduced X3MLBench, a benchmark of real-world, expert-validated X3ML mapping projects, and evaluated different LLMs and prompting strategies for X3ML mapping generation. The results showed that examples and domain-relevant context play a crucial role in the quality of the generated mappings.

Retrieval-Augmented Generation (RAG) [13, 14] provides one way to supply such context, by retrieving relevant external material and providing it to the LLM as part of the prompt. In schema matching and ontology matching, retrieval-enhanced methods have already shown that selecting relevant schema elements before prompting can improve both scalability and accuracy [15]. Related work on fine-grained retrieval over structured inputs [16] also supports the idea that raw structured representations may need to be transformed into more suitable retrieval units before being used as LLM context. For schema mapping generation, this principle can be applied by retrieving mapping examples from previous X3ML mappings and using them as contextual guidance for a new mapping task.

Our work builds on these lines of work but addresses a distinct gap. Existing studies show that LLMs can support knowledge graph construction and, in some cases, generate declarative mappings. However, they do not systematically compare how different forms of guidance affect the quality of generated mappings. This paper focuses on X3ML as a concrete declarative mapping language and evaluates five guidance strategies: syntactic guidance, ontology documentation, retrieval from mapping corpora, correspondence-enhanced retrieval, and expert rules. In this respect, the paper extends our previous work from feasibility and benchmark-based evaluation towards a more focused investigation of how contextual and expert knowledge affect the accuracy, semantic appropriateness, and practical usefulness of LLM-generated schema mappings.

3. Research Questions & Methodology

3.1. Research Questions

This paper addresses the following research questions:

- **RQ1: does guidance improve LLM-based schema mapping generation compared to a syntax-only baseline?** It examines whether the provision of additional information beyond syntactic structure leads to more accurate and useful mappings.
- **RQ2: which type of guidance is more effective for schema mapping generation?** It compares the different forms of guidance examined in this work, in order to understand whether mapping quality benefits more from explanatory ontology context, reference mappings, or direct expert knowledge.
- **RQ3: what types of errors are reduced or introduced by each guidance approach?** It focuses on the qualitative differences between the generated mappings. It investigates whether certain modes are more effective and in which parts of the process they perform better.
- **RQ4: Do the observed benefits of guidance remain consistent across different LLM families?** It examines whether the guidance strategies that improve X3ML mapping generation with one LLM show similar trends when applied with another one.

3.2. Methodology

The methodology investigates how different forms of guidance affect the ability of LLMs to generate X3ML schema mappings. For each mapping task, the system receives a common input consisting of the source data (or a small representative sample) and the target ontology. The main experimental variable is the type of guidance provided to the LLM during prompt construction.

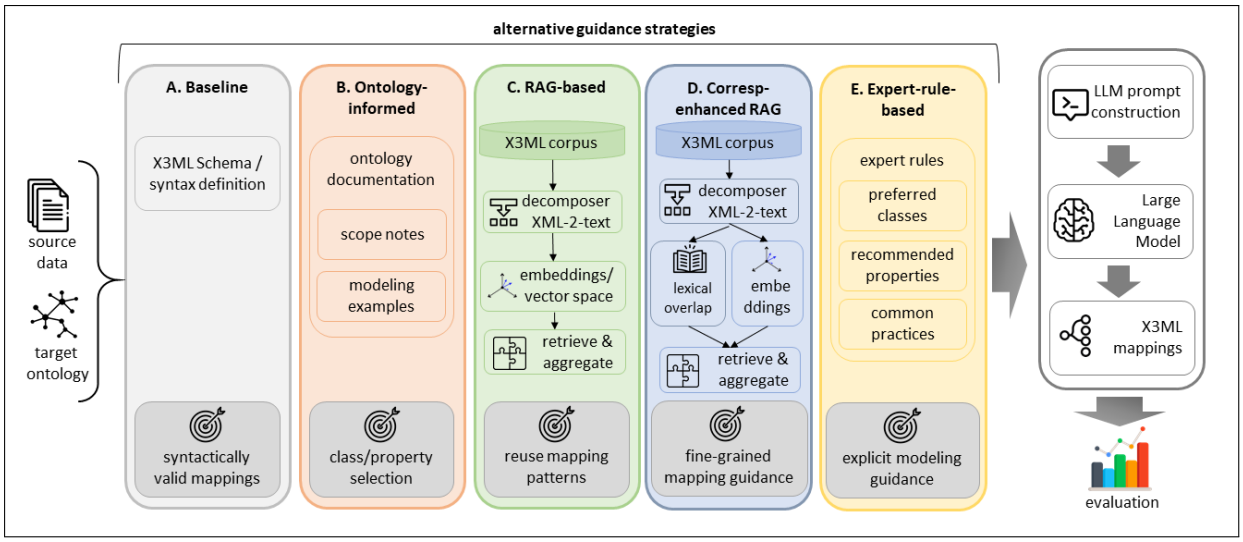


Figure 2: Overview of the proposed methodology and the five guidance strategies for LLM-based X3ML schema mapping generation

As shown in Figure 2, five alternative guidance strategies are examined: a baseline mode using only the syntactic definition of X3ML; an ontology-informed mode using target ontology documentation; a RAG-based mode that retrieves relevant X3ML mapping patterns for individual XML elements using embedding-based semantic similarity; a correspondence-enhanced RAG mode that re-ranks semantically retrieved examples using lexical overlap between XML elements; and an expert-rule-based mode using explicit modelling rules. In all cases, the selected guidance material is combined with the common mapping task input and used to construct the final prompt submitted to the LLM. The output of the LLM is an X3ML mapping, which is then evaluated through automatic comparison with reference mappings and analysis of recurring error patterns.

The ***baseline mode*** represents the minimum level of guidance required for the generation of X3ML mappings. The LLM is provided with the input mapping task and the syntactic structure of X3ML through the X3ML schema. This mode does not provide additional semantic information about how source elements should be interpreted or how target ontology entities should be selected. Therefore, it serves as a reference point for assessing the added value of richer guidance strategies.

The ***ontology-informed*** mode extends the baseline by providing documentation of the target ontology, including class and property descriptions, scope notes, usage guidelines, and modelling examples. This semantic context is intended to help the model avoid superficial name-based choices and select ontology classes and properties that better reflect the intended conceptualisation of the target ontology.

The ***RAG-based*** mode exploits a repository of previously created X3ML mappings at the level of individual XML elements rather than complete mapping projects. Each indexed entry consists of an XML element extracted from a previous source file, together with the X3ML mapping pattern that was used to represent it. During the mapping process, the input XML document is decomposed into individual elements and each element is used as an independent query for retrieving similar previously mapped elements. The X3ML patterns associated with the retrieved elements are then aggregated and provided to the LLM as contextual examples.

Before indexing and retrieval, source data are converted into textual representations. This preprocessing step is motivated by prior work on retrieval over structured data [16], which shows that raw structured representations may not provide optimal retrieval units. In our setting, raw XML fragments include markup, nesting, and compact labels that may be insufficiently informative when embedded directly. We therefore transform each XML element into a short textual narrative that preserves its structural context and intended meaning. For example, instead of embedding only a raw element such as `<date>`, the system represents it together with its parent and local context, such as "the date of birth". These textual narratives are then embedded and indexed, allowing the retrieval step to compare

semantically meaningful descriptions rather than raw XML fragments.

Let Γ denote the repository of indexed mapping examples, where each entry is represented as $\gamma = (s_\gamma, p_\gamma)$, where s_γ is the textual representation of an XML element extracted from a previous source file and p_γ is the associated X3ML mapping pattern. Given a new XML document q, the system extracts the set of XML elements $E(q) = \{e_1, e_2, \ldots, e_m\}$.

Each element $e_j \in E(q)$ is also transformed into a textual representation that preserves its local XML context. For each element, the embedding vector is computed and compared with the indexed elements using cosine similarity:

$$sim_{RAG}(e_j, s_\gamma) = \frac{emb(e_j) \cdot emb(s_\gamma)}{\|emb(e_j)\| \, \|emb(s_\gamma)\|},$$

where $emb(\cdot)$ denotes the embedding function applied to the textual representation of an XML element. The most similar mapping examples are retrieved and supplied to the LLM as contextual guidance for generating the mapping of e_j. This approach assumes that semantically similar XML elements are likely to share similar X3ML modelling patterns, even when they originate from different mapping projects.

The ***correspondence-enhanced RAG*** mode extends the previous retrieval strategy by incorporating lexical evidence into the ranking of retrieved examples. It uses the same element-level textual representations described above, but first retrieves an expanded set of candidate mapping examples using embedding-based similarity. These candidates are then re-ranked by combining their semantic similarity score with a lexical overlap score computed between the query element and each candidate example. The lexical score is computed over the enriched textual representations that include information such as the XML tag, value, attributes, parent context, and associated mapping content. In this way, the enhanced mode favours examples that are both semantically close in the embedding space and lexically aligned with the queried XML element. Formally, for each query element e_j, the re-ranking score is defined as:

$$score_{CE\text{-}RAG}(e_j, \gamma) = \alpha \cdot sim_{RAG}(e_j, s_\gamma) + (1 - \alpha) \cdot lex(e_j, \gamma),$$

where $lex(e_j, \gamma)$ is the lexical overlap score between the textual representation of the query element and the enriched textual representation of the indexed example. In our implementation, it is computed as the proportion of query tokens that also occur in the candidate representation: $lex(e_j, \gamma) = \frac{|tokens(e_j) \cap tokens(\gamma)|}{|tokens(e_j)|}$. The examples with the highest score are then selected as guidance for the LLM. In the experiments, α is kept fixed, giving slightly higher weight to embedding-based similarity than to lexical overlap.

This re-ranking strategy is intended to reduce ambiguities that may arise from purely semantic retrieval. For example, a generic element such as `<id>` may be semantically close to several identifier-related elements, such as `<Identifier>`, `<idno>`, or`<IdentifierNumber>`, even when they appear in different structural contexts or correspond to different mapping patterns. By incorporating lexical correspondences, the enhanced retrieval process gives higher priority to examples that more closely match the queried XML element, resulting in more precise guidance for the LLM.

The ***expert-rule-based*** mode provides the LLM with explicit modelling rules formulated by mapping experts. These rules are intended to guide the model in selecting appropriate ontology constructs and avoiding common modelling errors. They may describe preferred classes for specific types of source elements, recommended properties for expressing particular relationships, constraints on valid modelling patterns, or common practices followed during X3ML mapping construction. The rules used in the experiments were defined by mapping experts before the final evaluation and were kept fixed across all tasks and models. They express general modelling preferences and constraints rather than task-specific corrections to generated outputs. The complete rule set is available in the accompanying GitHub repository. Unlike the RAG-based mode, which relies on examples retrieved from previous projects, the expert-rule-based mode provides direct and explicit guidance. This allows expert knowledge to be expressed in a compact and controlled form. The expected benefit is that the LLM can use these rules to make more informed modelling decisions, especially in cases where the source schema is ambiguous

or where ontology documentation alone is not sufficient to determine the most appropriate mapping pattern. This mode reflects a practical human-in-the-loop scenario, where experts do not manually write the entire mapping from scratch, but provide rules and constraints that shape the output of LLMs.

For each mode, the LLM produces an X3ML mapping for the same mapping task. This design allows the outputs of the five guidance strategies to be compared under similar conditions. The generated mappings are first assessed through automatic evaluation, where they are compared against reference mappings using quantitative metrics that capture structural and semantic correspondence.

4. Experiments and Evaluation

4.1. Experimental Setup

Here we describe the main resources and configuration choices used in the experiments; the benchmark and mapping tasks used for evaluation (in §4.1.1), the corpus of existing X3ML projects used to support the RAG-based approach and the expert rules (in §4.1.2), the LLMs employed for mapping generation (in §4.1.3), and the implementation details of the generation workflow (in §4.1.4).

4.1.1. Benchmark and Mapping Tasks

The experiments were carried out using X3MLBench [5], a benchmark consisting of real-world, expert-validated X3ML mapping projects. Each benchmark item corresponds to a schema mapping task, where a structured XML source has to be mapped to a target ontology using X3ML. The mappings included in X3MLBench use CIDOC CRM as their target ontology and cover different types of entities, including persons, locations, events, and related contextual information. For each task, the manually constructed X3ML mapping is used as the gold standard mapping, denoted as M_{GS} in Section 4.2.

The benchmark includes mapping tasks with varying levels of source-data complexity, including sources with both flat and more sophisticated XML structures. For each task, the LLM received the same basic input and the target ontology, while additional guidance differed depending on the examined approach. This setup allows the generated mapping to be compared under common conditions and makes it possible to isolate the effect of each guidance strategy. The X3MLBench mappings and related material are available through the corresponding GitHub repository[3].

4.1.2. X3ML Mapping Corpus and Expert Rules

The retrieval-based approaches rely on a corpus of existing X3ML mapping projects collected from the 3M[4] installation. These projects correspond to real mapping activities carried out by human experts in data integration scenarios and therefore provide a useful source of reusable mapping knowledge. In contrast to synthetic examples, the corpus captures actual modelling decisions, recurring X3ML patterns, and practical uses of target ontologies in concrete transformation tasks.

Before being used in the experiments, the initial collection was curated in order to keep only the complete and valid X3ML mapping projects. This curation step removed testing, incomplete, invalid, and duplicate mappings, and also normalized projects with multiple input sources by merging them into a single input source representation. As a result, the corpus was reduced from approximately 800 initial X3ML projects to 338 curated projects. The majority of these mappings use CIDOC CRM and its extensions as target ontologies, reflecting the strong use of X3ML in cultural heritage integration scenarios. However, the corpus also includes mappings targeting other ontological models. This curated corpus is used as the common retrieval resource for the approaches that exploit previous mappings as guidance.

To reduce the risk of corpus/benchmark leakage, the projects for the evaluation tasks were excluded from the retrieval corpus before building the indexes used by the RAG-based modes. This prevents the

[3] https://github.com/ymark/x3ml_comparator
[4] https://demos.isl.ics.forth.gr/3m/

retrieval process from directly accessing the gold standard mapping used for evaluation. Although the corpus may still contain mappings from similar domains or involving similar ontology patterns, these are treated as legitimate reusable examples, since the aim of the retrieval-based modes is precisely to exploit recurring mapping knowledge across projects.

In addition to the retrieval corpus, the expert rules are treated as a separate experimental resource. They provide reusable guidance material in the form of modelling preferences and constraints, but are not derived from the generated outputs of the evaluated runs. The rules were defined before the final evaluation, kept fixed across tasks and models, and made available through the accompanying GitHub repository.

4.1.3. Large Language Models and Configuration

The experiments were conducted using LLMs accessed through two different providers: Mistral Medium 3.5[5] and OpenAI GPT-4o[6]. These models were selected in order to examine whether the proposed guidance strategies behave consistently across different LLM families and provider settings.

It should be noted that the objective of this work is not to benchmark or rank individual LLMs. Instead, the focus is on evaluating the effect of different guidance scenarios on schema mapping generation. For this reason, the same experimental protocol was followed for all examined modes: each model received the same mapping task input, while the additional guidance varied according to the experimental alternative. The generation parameters were kept stable across runs, where applicable, using temperature 0.8 and top-p 0.95. The maximum available context window differed by model, with Mistral Medium 3.5 supporting up to 256K tokens and GPT-4o supporting 128K tokens.

4.1.4. Implementation Details

The experimental pipeline was implemented in Python and was designed to support the execution of all experiments in an automated manner. The implementation handles the preparation of the input prompts, the execution of the different guidance strategies, the communication with the selected LLM APIs, the storage of the generated X3ML mappings and the organization of the outputs for subsequent evaluation. In addition, the pipeline supports the automatic comparison of generated mappings with the gold standard mappings, the computation of the evaluation metrics, and the generation of plots summarizing the performance of the different modes. This allows the complete set of experiments to be executed without user intervention, ensuring that the same procedure is consistently applied across all mapping tasks, models, and guidance strategies.

The implemented software does not require significant computational resources, since the most demanding generation step is performed through external LLM APIs. The experiments were carried out on a single laptop with standard specifications[7]. The corresponding GitHub[8] repository includes the code used for running the experiments, the necessary scripts, the configuration files, the expert rules used in the expert-rule-based mode, the generated outputs, and the detailed evaluation results.

For the retrieval-based modes, vector indexing is implemented using FAISS (Facebook AI Similarity Search) [17]. Source files and source elements were represented using the *multilingual-e5-base* embedding model [18]. In both RAG-based modes, the new source file is decomposed into elements, embedded using the same model and compared against the corresponding FAISS index in order to retrieve the most relevant mapping examples to use in the prompt. For the correspondence-enhanced RAG mode, the re-ranking score was instantiated with $\alpha = 0.55$, resulting in a weighted combination of 0.55 for the embedding-based similarity score and 0.45 for the lexical overlap score.

[5] https://huggingface.co/mistralai/Mistral-Medium-3.5-128B
[6] https://developers.openai.com/api/docs/models/gpt-4o
[7] Ryzen 5 7430U, 16 Gb RAM
[8] https://github.com/angelybuckel29-code/xmate_bench_public

4.2. Evaluation Metrics

Formally, let S denote a source schema and T denote a target ontology. Since X3ML focuses on hierarchical formats as source schemata, such as XML documents, source elements are represented through source paths. We use $n(S)$ to denote the set of source nodes/paths. Similarly, we use $n(T)$ to denote the set of target ontology nodes (i.e. classes or value nodes), and $r(T)$ to denote the set of target ontology relations (i.e. properties).

An X3ML mapping M consists of a set of mapping rules $M = \{m_1, m_2, \ldots, m_k\}$. Each mapping rule ($m_i$) consists of a domain/node correspondence and a set of link correspondences, $m_i = (dm_i, L_i)$. The domain correspondence (dm_i) associates a source/node path with a target ontology node, $dm_i = (s_i, t_i), \quad s_i \in n(S), \quad t_i \in n(T)$. This correspondence captures the main source context of the mapping rule and the target entity that is created or described by that rule. The set containing the links of the mapping rule is defined as: $L_i = \{l_{i1}, l_{i2}, \ldots, l_{ip}\}$, where each link associates a source node with a target relation-node pair, $l_{ij} = (s_j, (r_j, t_j))$, where $s_j \in n(S), \quad r_j \in r(T), \quad t_j \in n(T)$. The source part ($s_j$) identifies a node from the source, while the target pair ((r_j, t_j)) identifies the ontology property and target node to represent this information in the RDF graph. Eventually, all the target pairs, identified by the links of a mapping rule, will be linked with the target node, identified in the domain of the mapping (i.e., t_i). This reflects the structure of X3ML mappings, where a domain node is mapped to a target entity and the links specify how related source elements are transformed into ontology properties and connected entities.

For evaluation purposes, we define the content of a mapping rule as the union of its domain correspondence and link correspondences, $cont(m_i) = \{dm_i \cup L_i\}$. Accordingly, the content of a mapping project is defined as the union of the contents of all its mapping rules, $cont(M) = \bigcup_{i=1}^{k} cont(m_i)$. Let M_{GS} denote the gold standard mapping project for a given mapping task, and let M_{gen} denote the corresponding mapping generated by an LLM-based approach. Using the content function defined above, we compare the two mappings through the sets of correspondences they contain. More specifically, a correspondence in the generated schema mapping is considered correct when it is also included in the gold standard. Based on this set-based comparison, we define precision, recall and F1-Score as follows.

Precision measures the proportion of correct correspondences generated: $Prec(M_{gen}, M_{GS}) = \frac{|cont(M_{gen}) \cap cont(M_{GS})|}{|cont(M_{gen})|}$.

Recall measures the proportion of gold standard correspondences that were successfully generated: $Rec(M_{gen}, M_{GS}) = \frac{|cont(M_{gen}) \cap cont(M_{GS})|}{|cont(M_{GS})|}$.

The F1-score is the harmonic mean of precision and recall, providing a simple measure that balances both correctness and completeness: $F1(M_{gen}, M_{GS}) = 2 \cdot \frac{Prec(M_{gen}, M_{GS}) \cdot Rec(M_{gen}, M_{GS})}{Prec(M_{gen}, M_{GS}) + Rec(M_{gen}, M_{GS})}$.

This set-based evaluation provides a controlled and reproducible basis for comparing the examined guidance strategies. However, it does not capture all cases of partial correctness or semantically equivalent alternative mappings. For example, a generated mapping may follow a different but still meaningful ontology path, or may be partially correct while failing to exactly match the gold standard correspondence. Therefore, the reported scores should be interpreted as a strict comparison against the reference mappings.

4.3. Evaluation Results

Figure 3 reports the F1-score obtained by each mode for each individual mapping task, while Table 1 provides the corresponding numerical values. In general, the baseline mode consistently achieves the lowest or among the lowest scores, confirming that the syntactic structure of X3ML alone is not sufficient to produce accurate mappings. Adding ontology documentation improves results in several tasks, indicating that explanatory information about the target ontology helps the model select more appropriate classes and properties. The expert-rule-based mode also improves over the baseline in most cases, although its effectiveness varies depending on whether the provided rules are directly relevant to the mapping task.

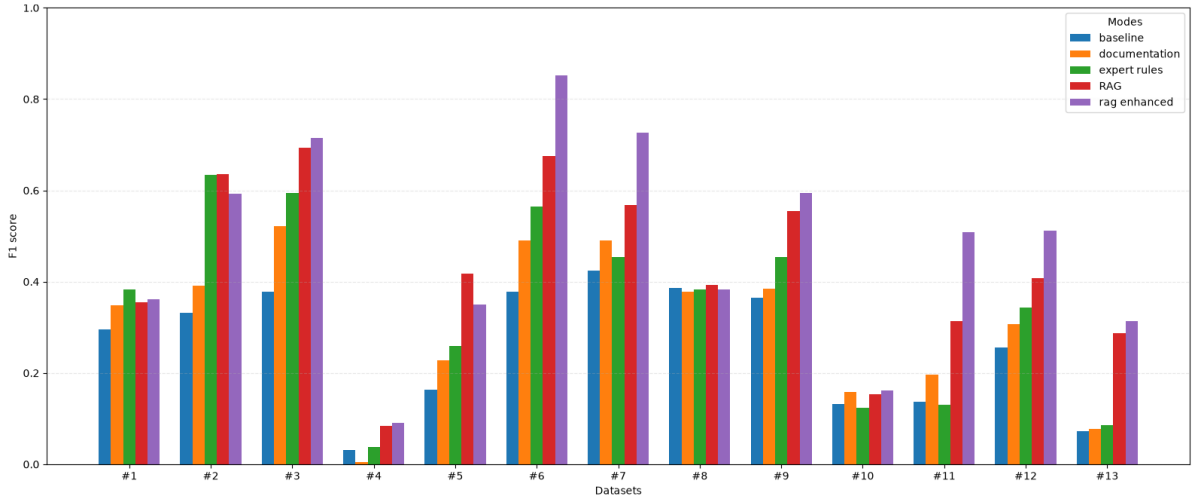


Figure 3: F1-score obtained for each X3MLBench mapping task by guidance strategy

Task	baseline	documentation	expert rules	RAG	RAG-enhanced
#1	0.295	0.348	**0.382**	0.355	0.361
#2	0.332	0.392	0.633	**0.635**	0.593
#3	0.378	0.522	0.595	0.694	**0.714**
#4	0.032	0.005	0.037	0.085	**0.091**
#5	0.163	0.227	0.259	**0.418**	0.350
#6	0.379	0.490	0.564	0.675	**0.852**
#7	0.424	0.490	0.455	0.568	**0.727**
#8	0.386	0.378	0.384	**0.392**	0.383
#9	0.364	0.384	0.453	0.555	**0.594**
#10	0.133	0.159	0.123	0.153	**0.162**
#11	0.138	0.196	0.131	0.314	**0.509**
#12	0.257	0.308	0.343	0.408	**0.512**
#13	0.072	0.078	0.085	0.287	**0.313**
Mean F1-Score	**0.258**	**0.306**	**0.342**	**0.426**	**0.474**

Table 1
F1-score per X3MLBench mapping task and guidance strategy

The per-task results depicted in Figure 3, show that retrieval-based guidance is useful in mapping scenarios where relevant previous examples are available. The standard RAG-based mode improves performance by retrieving previously mapped XML elements that are semantically similar to the query element and providing their associated X3ML mappings as contextual examples. The correspondence-enhanced RAG mode provides a more targeted form of guidance by re-ranking semantically retrieved examples according to lexical overlap with the queried XML element. This difference is particularly evident for ambiguous XML elements. For instance, the element `<id>` is represented only 48 times in the RAG knowledge base, while 269 related elements correspond to identifiers but use different tag names, such as `<identifier>`, `<IdentificationNumber>`, and `<idno>`. Since embedding-based retrieval is driven by semantic similarity, it may retrieve mappings associated with identifier concepts without sufficiently prioritizing the exact XML element or its local context. The lexical re-ranking step addresses this limitation by assigning higher scores to candidates that are both semantically similar and lexically aligned with the query element.

Figure 4 summarizes these results by showing the minimum, maximum, and mean F1-score for each guidance strategy. The aggregate view confirms the trend observed in the per-task analysis, with correspondence-enhanced RAG achieving the highest mean F1-score. It is also worth noting that the expert-rule-based and retrieval-based modes maintain a non-zero performance floor across all tasks, with the RAG-based modes achieving higher minimum scores than the baseline and ontology-documentation

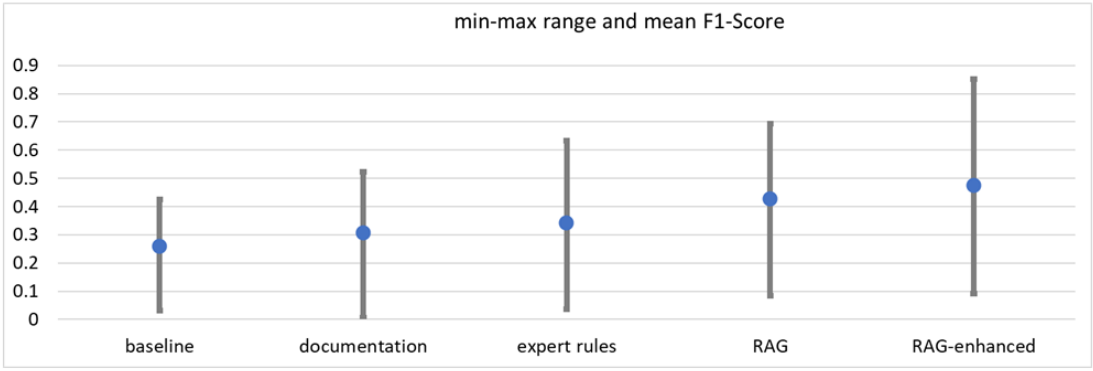


Figure 4: Minimum, maximum, and mean F1-score across X3MLBench mapping tasks for each guidance strategy

modes. This suggests that explicit expert knowledge and reusable mapping examples can provide useful fallback guidance even in difficult mapping scenarios. At the same time, the wide min-max ranges show that performance remains task-dependent, suggesting that the effectiveness of each guidance strategy is influenced by source-file complexity, mapping-pattern complexity, and the availability of relevant examples in the corpus.

To examine whether the observed benefits of guidance are specific to the selected LLM or generalize across different model families, we also repeated the experiments with a commercial LLM accessed through the OpenAI API. The purpose of this additional evaluation was not to compare LLMs, but to assess whether the relative effectiveness of the guidance strategies remains consistent when using a different model. Due to space limitations, we report here only the aggregate result, while the detailed per-task results are available in the accompanying GitHub repository. The results follow the same overall trend as those obtained with Mistral: both models achieved their best performance using the correspondence-enhanced RAG mode; Mistral achieved a mean F1-score of 0.474, while the OpenAI model achieved a mean F1-score of 0.418. Although there are differences in absolute performance, the results indicate that retrieval-based guidance, and especially correspondence-level guidance, remains beneficial across different LLM settings. This supports the conclusion that the proposed guidance mechanisms are not tied to a single model, while broader multi-model evaluation remains part of future work.

The results address the research questions as follows. For RQ1, all guidance strategies improve over the syntax-only baseline in terms of mean F1-score. For RQ2, retrieval-based guidance is the most effective among the examined alternatives, with correspondence-enhanced RAG achieving the highest mean performance. For RQ3, the error analysis shows that guidance reduces some structural and semantic failures, but does not eliminate malformed X3ML structures, incorrect links, or hallucinated mapping content. For RQ4, the GPT-4o experiments suggest that the benefits of correspondence-enhanced retrieval are not limited to a single LLM setting.

The results also relate to interpretability. Since the output is an explicit X3ML mapping, rather than a directly generated RDF graph, modelling decisions can be inspected, validated, and revised before the actual data transformation. The retrieval-based mode expose the modelling examples used as guidance, while the expert-rule-based mode makes the modelling constraints explicit. This improves traceability compared to prompt-to-RDF generation, although human inspection remains necessary.

4.4. Error Analysis and Common Failures

Beyond the quantitative evaluation, we inspected the generated mappings in order to identify recurring errors. This analysis helps explain the differences between the guidance strategies and provides a qualitative view of the limitations of LLM-based schema mapping generation. The observed errors can be grouped into three main categories. The first concerns structural errors, including malformed XML, incorrect formatting, missing or misplaced X3ML elements, malformed metadata sections, incorrectly

generated `<additional>` elements, and unclosed tags. Such errors prevented the generated mappings from being directly executable, even when part of the intended mapping was captured. Structural errors were the most frequent type of failure during the initial experimental runs. They were therefore used as feedback for refining the prompt templates, for example by adding explicit instructions to close all X3ML elements and preserve the required nesting of mapping and link constructs. This reduced many malformed-output cases, although some structural errors still remained in the final outputs.

The second category concerns missing or incorrect link definitions. In some cases, the model identified a relevant source element, but failed to create the corresponding X3ML link. In other cases, it created a link but associated it with an inappropriate target-side path, such as a wrong CIDOC CRM property, an unsuitable target class, or an incomplete class-property pattern. These errors are important because they may produce syntactically valid mappings that nevertheless express an incorrect semantic interpretation of the source data. The third category concerns hallucinations, where the model introduces ontology classes, properties, source paths, or X3ML structures that are not justified by the source data, the target ontology, or the retrieved guidance. Such errors include the generation of unsupported mapping structures or invented ontology terms. Finally, some failures were related to input complexity, especially when large XML inputs made it harder for the model to preserve all relevant source paths and generate complete mappings.

5. Conclusions and Future Work

This study highlights the practical value of guidance for producing more accurate and reusable LLM-generated schema mappings. Building on previous work on the difficulty of schema mapping construction and the limitations of weakly guided LLM generation, this paper compared five guidance strategies for X3ML mapping generation: syntactic, ontological, retrieval-based, correspondence-enhanced, and expert-provided guidance. The results show that the syntax-only baseline achieves the lowest performance, while the provision of ontology documentation and expert rules improve it by adding semantic and modelling context. The strongest results are obtained by the correspondence-enhanced RAG mode, indicating that combining embedding-based semantic retrieval with lexical re-ranking provides more effective guidance than semantic retrieval alone. Additional experiments with a commercial LLM further suggest that the benefits of guidance are not limited to a single model, but can generalize across different LLMs.

Future work will examine whether the proposed guidance strategies transfer beyond X3ML and CIDOC CRM-oriented cultural heritage scenarios to other mapping languages, source formats, domain, and ontologies. We will also investigate iterative self-verification mechanisms for detecting and revising hallucinated ontology terms, invalid X3ML structures, and missing correspondences. Further work will study the tuning of the correspondence-enhanced retrieval parameter α, explore more fine-grained evaluation methods for partial or semantically equivalent mappings, and use the observed error patterns to design validation rules, post-processing steps, and interactive human-in-the-loop refinement mechanisms.

Acknowledgments

We would like to thank Athina Kritsotaki from FORTH-ICS for reviewing the expert rules.

Declaration on Generative AI

The authors did not use generative AI systems for generating scientific content, analysis, results, figures or conclusions presented in the paper; such tools were used exclusively for language refinement and proofreading.

References

[1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, arXiv preprint arXiv:2303.18223 1 (2023) 1–124.

[2] Y. Marketakis, Y. Tzitzikas, Beyond prompt-to-rdf: A vision for scalable and explainable graph transformations via llm-assisted schema mappings, in: Proceedings of the Workshops of the EDBT/ICDT 2026 Joint Conference, 2026.

[3] Y. Marketakis, N. Minadakis, H. Kondylakis, K. Konsolaki, G. Samaritakis, M. Theodoridou, G. Flouris, M. Doerr, X3ml mapping framework for information integration in cultural heritage and beyond, International Journal on Digital Libraries 18 (2017) 301–319.

[4] M. Doerr, The cidoc crm, an ontological approach to schema heterogeneity, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2005.

[5] Y. Marketakis, M. Lintanff-Castel, Y. Tzitzikas, Using llms to automate schema mappings for rdf knowledge graphs construction, in: Proceedings of the 41st ACM/SIGAPP Symposium on Applied Computing, SAC '26, Association for Computing Machinery, New York, NY, USA, 2026, pp. 1999–2000. URL: https://doi.org/10.1145/3748522.3779763. doi:10.1145/3748522.3779763.

[6] W. W. W. Consortium, et al., R2rml: Rdb to rdf mapping language (2012).

[7] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, R. Van de Walle, Rml: A generic language for integrated rdf mappings of heterogeneous data., Ldow 1184 (2014).

[8] P. Heyvaert, B. De Meester, A. Dimou, R. Verborgh, Declarative rules for linked data generation at your fingertips!, in: The Semantic Web: ESWC 2018 Satellite Events: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers 15, Springer, 2018, pp. 213–217.

[9] H. Bian, Llm-empowered knowledge graph construction: A survey, arXiv preprint arXiv:2510.20345 (2025).

[10] W. J. Schmidt, I. Grangel-González, T. Huschle, L. Wagner, E. Kharlamov, A. Paschke, Llm-supported mapping generation for semantic manufacturing treasure hunting, in: European Semantic Web Conference, Springer, 2025, pp. 84–101.

[11] M. Hofer, J. Frey, E. Rahm, Towards self-configuring knowledge graph construction pipelines using llms-a case study with rml, in: Fifth International Workshop on Knowledge Graph Construction@ ESWC2024, volume 3718, 2024.

[12] S. Karkee, E. Botoeva, S. Coombes, A. Jordanous, Ö. Kafalı, D. Lanti, Accessing semi-structured data with rml and llms, CEUR Workshop Proceedings, 2025.

[13] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, arXiv preprint arXiv:2312.10997 (2023).

[14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in neural information processing systems 33 (2020) 9459–9474.

[15] E. Sheetrit, M. Brief, M. Mishaeli, O. Elisha, Rematch: Retrieval enhanced schema matching with llms, arXiv preprint arXiv:2403.01567 (2024).

[16] Z. Guo, W. Geng, R. Mao, Ft-rag: A fine-grained retrieval-augmented generation framework for complex table reasoning, arXiv preprint arXiv:2605.01495 (2026).

[17] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with gpus, IEEE Transactions on Big Data 7 (2021) 535–547. doi:10.1109/TBDATA.2019.2921572.

[18] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual e5 text embeddings: A technical report, arXiv preprint arXiv:2402.05672 (2024).