On the Use of Large Language Models for Schema Mapping Generation in Ontology-Based Data Integration

Yannis Marketakis
marketak@ics.forth.gr
Institute of Computer Science, FORTH-ICS
Computer Science Department, University of Crete
Heraklion, Crete, Greece

Milio Lintanff–Castel
milio.lintanff--castel@etudiant.univ-rennes.fr
Lannion Institute of Technology, University of Rennes
Rennes, Upper Brittany, France

Yannis Tzitzikas
tzitzik@ics.forth.gr
Institute of Computer Science, FORTH-ICS
Computer Science Department, University of Crete
Heraklion, Crete, Greece

Abstract

Schema mapping is a central task in ontology-based data integration, enabling the transformation of heterogeneous data sources into semantically interoperable Knowledge Graphs. Despite its importance, the construction of schema mappings remains a complex and expert-driven process, requiring both domain knowledge and familiarity with formal mapping languages. In this paper, we investigate the use of Large Language Models (LLMs) for schema mapping generation, framing it as a constrained structured generation problem that requires adherence to strict syntactic, semantic, and ontological constraints. Focusing on the X3ML mapping language and the use of CIDOC CRM ontology in the cultural domain, we introduce X3MLBench, a curated benchmark of real-world, expert-validated mapping projects, along with an automated evaluation framework for assessing mapping quality across multiple dimensions. We conduct a systematic empirical study of several state-of-the-art LLMs under different prompting strategies, isolating the role of structural guidance and contextual information in generating valid mappings. Our results show that while LLMs can produce syntactically valid and partially accurate mappings, their performance depends critically on the availability of structured examples and domain-relevant context. Beyond quantitative evaluation, we analyze recurring error patterns, revealing limitations in ontology alignment and structural consistency when input data are ambiguous or weakly structured. These findings highlight both the potential and the limitations of LLMs in structured semantic generation tasks, and motivate semi-automated workflows where LLMs assist experts by generating initial mapping specifications that are subsequently validated and refined.

ACM Reference Format:
Yannis Marketakis, Milio Lintanff–Castel, and Yannis Tzitzikas. 2026. On the Use of Large Language Models for Schema Mapping Generation in Ontology-Based Data Integration. In *Proceedings of The 14th EETN Conference on Artificial Intelligence (SETN 2026).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

SETN 2026, September 9–11, 2026, Chania, Greece

ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Large Language Models (LLMs) [38] have recently demonstrated strong capabilities in generating structured outputs across a wide range of tasks, including code generation, data transformation, and knowledge extraction. However, their ability to produce outputs that satisfy strict formal constraints, such as those imposed by schema mapping languages and ontologies, remains limited and insufficiently understood. Tasks that require precise alignment between heterogeneous data schemas and formal ontological models pose particular challenges, as they demand not only linguistic competence but also structural consistency and semantic correctness.

Schema mapping is a fundamental component of ontology-based data integration [4, 22], enabling the transformation of heterogeneous data sources into semantically enriched and interoperable Knowledge Graphs. In this setting, mappings define explicit correspondences between elements of a source schema and concepts in a target ontology, effectively specifying the transformation logic from raw data to structured semantic representations. Despite their importance, schema mappings are typically created manually, requiring significant effort from domain experts and ontology engineers, as well as deep familiarity with both the source data and the target ontology.

Recent work has explored the use of LLMs for transforming structured data directly into RDF representations [21, 39, 35]. While these approaches demonstrate the expressive power of LLMs, they typically treat the transformation process as a black-box generation task, lacking an explicit and reusable representation of the transformation logic. In contrast, schema mappings provide a declarative, interpretable, and reusable specification of data transformations, enabling validation, debugging, and control over the integration process. This distinction motivates a shift from end-to-end generation toward LLM-assisted construction of explicit mapping specifications, aligning with broader goals in interpretable and controllable AI.

In this paper, we investigate the use of LLMs for schema mapping generation, focusing on mappings expressed in the X3ML language [19] and targeting the ISO 21127:2023[1] CIDOC-CRM ontology [6]. We selected this ontology since it is widely used in the cultural domain, and there are hundreds of human-made mappings towards that ontology. We formulate schema mapping generation as a constrained structured generation problem, where outputs must simultaneously satisfy syntactic validity, semantic alignment,

[1] https://www.iso.org/standard/85100.html

and structural consistency. To support systematic evaluation, we introduce a benchmark of real-world mapping projects and conduct an extensive empirical study across multiple LLMs and prompting strategies, providing insights into the capabilities and limitations of current models in structured semantic generation tasks.

Main Contributions. In this work, we study schema mapping generation as a structured semantic generation problem under strict syntactic and ontological constraints, and investigate the extent to which Large Language Models (LLMs) can effectively support this task. Our main contributions are as follows:

(1) **Problem formulation**. We formalize schema mapping generation as a constrained structured output task, where the goal is to produce mappings that simultaneously satisfy: (i) syntactic validity with respect to the X3ML specification, (ii) semantic alignment with a target ontology (CIDOC CRM), and (iii) structural consistency across interconnected mapping components. This formulation highlights the challenges of applying LLMs in settings that require both symbolic correctness and semantic reasoning.
(2) **Benchmark dataset and evaluation framework**. We introduce a curated benchmark composed of real-world, expert-validated schema mapping projects from the cultural heritage domain. The benchmark is accompanied by an automated evaluation framework that enables systematic comparison between generated and gold-standard mappings across multiple dimensions of quality, including syntactic validity, semantic correctness, and structural correspondence.
(3) **Systematic empirical study of LLMs**. We conduct a controlled and extensive empirical evaluation of multiple state-of-the-art LLMs under different prompting strategies (zero-shot, syntax-aware, and in-context). Our study isolates the effect of structural guidance and contextual information, providing a detailed analysis of how these factors influence the ability of LLMs to generate valid and meaningful schema mappings.
(4) **Quantitative and qualitative analysis of model behavior**. Beyond aggregate performance metrics, we provide an in-depth analysis of LLM outputs, identifying recurring error patterns related to ontology misuse, structural inconsistencies, and invalid mappings. This analysis offers insights into the limitations of current LLMs in tasks that require precise semantic alignment and adherence to formal constraints.
(5) **Insights for semi-automated data integration workflows.** Based on our findings, we outline the role of LLMs as assistive tools in schema mapping workflows, where they can generate initial mapping specifications that are subsequently validated and refined by domain experts. Our results highlight both the potential and the limitations of LLMs in supporting interpretable and controllable semantic data integration.

To the best of our knowledge, this is the first systematic study of LLMs for schema mapping generation under explicit structural and ontological constraints using real-world X3ML mappings.

Summary and scope of this work. Overall, this work aims to bridge the gap between the expressive capabilities of LLMs and the strict requirements of formal data integration workflows. By treating schema mapping generation as a constrained structured task and evaluating LLMs within a realistic, ontology-driven setting, we provide a clearer understanding of when and how such models can be effectively used. Our findings suggest that, while LLMs are not yet reliable as fully autonomous mapping generators, they can serve as powerful assistive tools when combined with appropriate structural guidance and human oversight. This positions LLMs as a key component in emerging semi-automated and human-in-the-loop data integration pipelines, where interpretability, controllability, and correctness remain essential.

2 Related Work

Recent work has explored multiple directions for transforming structured and semi-structured data into Knowledge Graphs, including both traditional schema-based approaches and emerging LLM-driven approaches. The construction of RDF graphs, as the result of data transformation from heterogeneous data sources has already been studied in the literature. Existing approaches span a wide range of techniques, including schema-based mappings, declarative transformation languages, and large-scale semantic integration frameworks. Surveys such as [34, 26] provide overviews of methods for generating RDF graphs from structured or semi-structured data resources. [22] reviews approaches for large-scale linked data integration, focusing on architectures, tools, and services that support interoperability across heterogeneous sources.

A central component of semantic data integration is schema and ontology matching. This problem has been studied extensively in [29], which proposes a classification framework for understanding and comparing different matching techniques. Similarly, [25] describes a taxonomy approach that achieves a partial automation of the schema matching for specific application domains. [1] analyzes existing efforts to benchmark schema matching and mapping tools, aiming to establish standard comparison criteria that help users, developers, and researchers evaluate and improve these systems. These works highlight the complexity of aligning heterogeneous schemas and the need for scalable and reliable mapping solutions. Such works on schema and ontology matching systems are relevant to our work as they address the identification of correspondences between schema entities; however, they do not directly generate executable X3ML mapping specifications, including target paths and structural links.

Moreover, there has been a growing interest in leveraging LLMs for tasks related to Knowledge Graph construction and data transformation. Several works explore the use of LLMs for generating RDF representations directly from input data. For instance, LLM2KB [23] proposes a system for constructing knowledge bases using instruction-tuned LLMs, while [3] introduces an iterative prompting approach for Knowledge Graph construction. Other approaches, i.e. [9], evaluate the ability of LLMs to generate RDF graphs in specific serializations, such as Turtle syntax, or apply LLMs to domain-specific transformations tasks, including relational data conversion for the energy domain (i.e. SQLMorpher [28]), and general-purpose data transformation workflows (i.e. [11]). In addition, LLMs have been used to support tasks related to Knowledge Graph construction, such as entity resolution [16], entity alignment [37], and ontology population [24].

However, most of these approaches rely on directly generating RDF outputs from input data, effectively treating the transformation as an end-to-end process. As a result, they lack an explicit intermediate representation of the transformation logic, limiting their interoperability, controllability, and reusability. These limitations have been explicitly highlighted in recent work, which argues for repositioning LLMs from black-box data transformers to assistive generators of schema mappings, enabling more transparent and maintainable transformation workflows [20]. In contrast, schema mappings provide a declarative and interoperable specification of how source data are transformed into ontology-based representations, enabling validation, debugging, and reuse of the transformation process.

The works most closely related to ours focus on the use of LLMs for schema mapping generation and assistance. In [14], the authors investigate the automatic generation of schema mappings using RML [5] for semi-structured data, evaluating LLMs on their ability to produce mappings based on a small ontology derived from DBpedia resources. Similarly, [2] approaches LLMs as schema-mapping assistants in dynamic environments with evolving data sources, highlighting challenges such as output instability, the complexity of expressive mappings, and the cost associated with repeated model invocations. Another closely related work is [27], which explores the generation of YARRRML [12] schema mappings in a manufacturing context, proposing techniques such as ontology size management and contextual reduction to address token limitations, and evaluating results against manually created mappings and expert feedback. These works are the closest to ours, as they also investigate LLM-assisted generation of declarative mapping specifications. However, they target different mapping languages and settings, such as RML and YARRRML, whereas our work focuses on X3ML mappings against real-world CIDOC CRM mapping projects and evaluates additional X3ML-specific requirements including syntactic validity, target-resource correctness, and target-path connectivity.

Positioning of our work. While prior work has demonstrated the potential of LLMs in knowledge graph construction and schema mapping tasks, existing approaches remain limited in at least one of the following aspects: (i) they treat transformation as an end-to-end generation problem without producing explicit and reusable mapping specifications, (ii) they focus on simplified or synthetic settings with limited evaluation scope, or (iii) they lack systematic analysis of how LLM behavior is affected by structural constraints and prompting strategies.

In contrast, our work explicitly frames schema mapping generation as a constrained structured generation problem, requiring simultaneous satisfaction of syntactic, semantic, and ontological constraints. Furthermore, we introduce a curated benchmark of real-world, expert-validated mappings expressed in X3ML, enabling a more realistic and rigorous evaluation setting. Finally, beyond reporting aggregate performance, we provide a systematic empirical and qualitative analysis of LLM behavior under different prompting strategies, offering insights into their strengths and limitations in structured semantic generation tasks.

3 Background: The X3ML Framework

The X3ML framework [19] provides the resources and technology for transforming structured data into RDF [18]. One of the distinctive characteristics of X3ML is that it decouples the schema mapping definition from the generation of URIs. This separation allows domain and ontology experts to focus on defining the schema mappings (i.e. how the source fields are mapped to ontology classes and properties), while technical experts handle the generation of URIs. It includes the *X3ML language*, a declarative language and human-readable language for defining schema mappings. Mappings are described in terms of domains and links and can be reviewed and discussed by non-technical people as well. Moreover, X3ML language supports rich mapping constructs (e.g. intermediate or additional nodes) for constructing multi-step target paths, all without writing any code. This is particularly important for CIDOC CRM, where a simple source relation may correspond to a multi-hop target pattern. For instance, a relation between a painting and its painter can be represented through an intermediate `E12_Production_Event` connected to the `E21_Person` who carried it out. X3ML supports such cases through target paths with multiple nodes and relationships within the same mapping link.

Figure 1 demonstrates some indicative X3ML mappings (part C) for transforming some source data (part A) as ontology instances (part B) with respect to CIDOC CRM. More specifically, the schema mappings describe the transformation of all `painting` elements from the source schema as instances of the class `E22_Human-Made Object` (as shown in *Domain* part), and subsequently, *Links* are used to connect these instances with additional information (e.g. title, painter, etc.) using the appropriate properties and classes. The illustrated schema mappings are designed using 3M Editor[2], a user-friendly web application that is used for defining and managing X3ML mappings.

Based on the described schema mappings, users can specify how the URIs of the transformed data will be formulated. This is achieved through the *generator policy*, which uses patterns to define how URIs and RDFS labels should be structured, using parameters derived from source data or constant values. These parameters can be populated with values extracted from the original data sources or assigned fixed constants. Both schema mappings and generator policy descriptions can be authored using tools such as the 3M Editor (i.e. shown in Figure 1), and exported as XML files conforming to X3ML XSD schema.

Regarding supported formats, X3ML mappings and generator policies are encoded as XML documents conforming to the X3ML schema. In this work, the source records are provided in XML, which is the native input format of the *X3ML Engine*; other structured formats, such as CSV, JSON, or relational data, can be used only after being transformed into an XML representation and are outside the scope of the present benchmark. The target model is referenced through URI-based ontology terms. Although this paper focused on CIDOC CRM, X3ML mappings can refer to classes and properties from other ontologies and vocabularies through proper namespace prefixes.

[2] https://demos.isl.ics.forth.gr/3m/

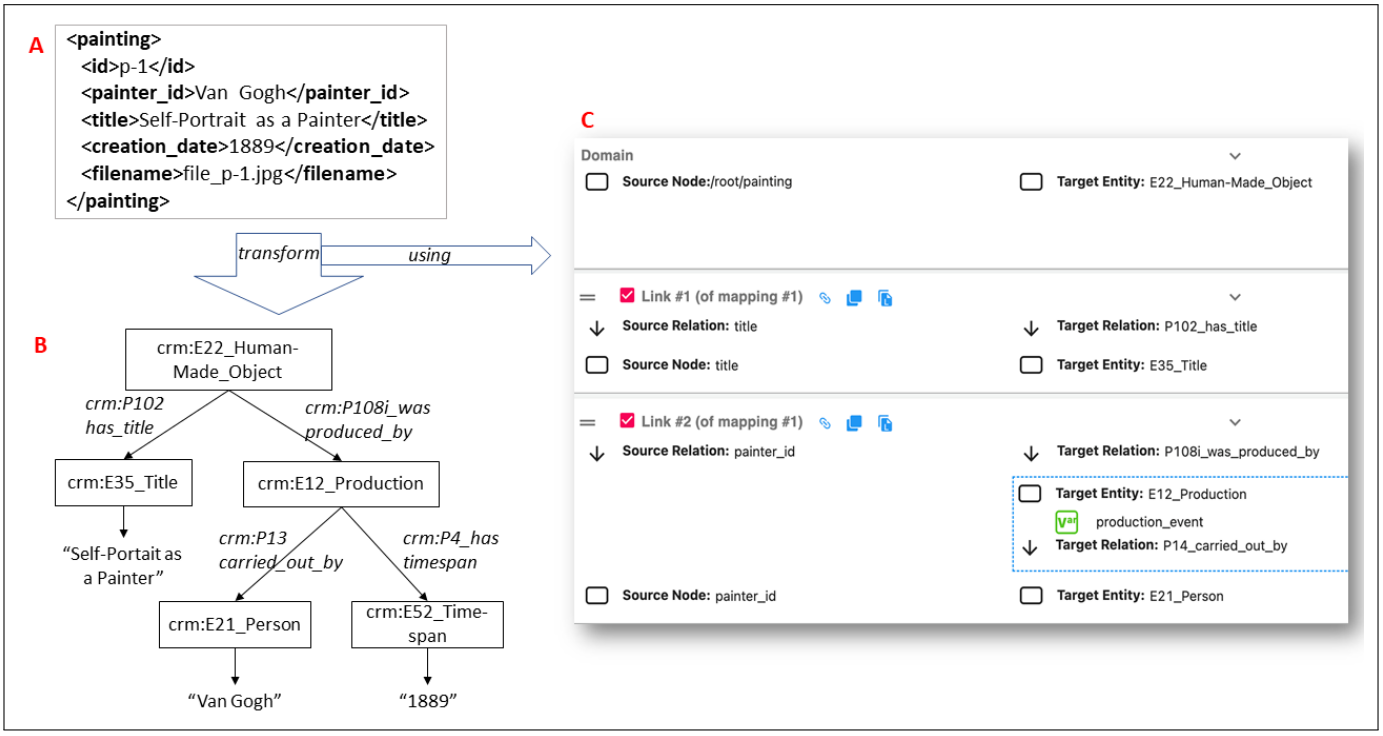

Figure 1: Demonstration of the schema mappings for transforming a source dataset (part A), into RDF instances with respect to CIDOC-CRM (part B), using X3ML definitions (part C)

After the mappings have been defined, the process continues with the transformation of data resources. This is carried out using *X3ML Engine*, a Java-based tool that executes the mappings to produce RDF output. It takes as input: (a) the source data expressed in a structured manner (i.e. XML), (b) the X3ML mapping definition files, and (c) the generator policy definitions; It traverses the elements of the source data, applies the mapping rules defined in the X3ML files, and generates the corresponding RDF triples. Below, we provide an indicative example of the XML representation of the schema mappings in X3ML for the example shown in Figure 1. For example, the generator `LocalTermURI` used in the mapping below, may define a URI pattern such as `http://example.org/resource/hierarchy/term`. If the mapping passes the constant value `painting` as `hierarchy`, and the XPath value `id/text()` as `term`, then a source painting with id 123, would be assigned the URI `http://example.org/resource/painting/123`. Thus, the X3ML mapping specifies which source values are passed to a generator, while the generator policy specifies how these values are converted into URIs or literals.

```
<x3ml>
  <mappings>
    <mapping>
      <domain>
        <source_node>/painting</source_node>
        <target_node>
          <entity>
            <type>crm:E22_Human-Made_Object</type>
            <instance_generator name="LocalTermURI">
              <arg name="hierarchy" type="constant">painting</arg>
              <arg name="term" type="xpath">id/text()</arg>
            </instance_generator>
            <label_generator name="CompositeLabel">
              <arg name="term1" type="constant">Painting with title:</arg>
              <arg name="term2" type="xpath">title/text()</arg>
            </label_generator>
          </entity>
        </target_node>
      </domain>
      <link>
        <path>
          <source_relation>
            <relation>title</relation>
          </source_relation>
          <target_relation>
            <relationship>crm:P102_has_title</relationship>
          </target_relation>
        </path>
        <range>
          <source_node>title</source_node>
          <target_node>
            <entity>
              <type>crm:E35_Title</type>
              <instance_generator name="LocalTermURI">
                <arg name="hierarchy" type="constant">title</arg>
                <arg name="term" type="xpath">text()</arg>
              </instance_generator>
              <label_generator name="Literal">
                <arg name="text" type="xpath">text()</arg>
              </label_generator>
            </entity>
          </target_node>
        </range>
      </link>
    </mapping>
  </mappings>
</x3ml>
```

4 The Benchmark X3MLBench

4.1 Collection of Mappings.

We use a collection of schema mappings designed and stored in the various 3M Editor deployments hosted by collaborating institutions. This collection comprises schema mapping projects developed for real-world use cases and datasets. All projects have been implemented and validated by domain and mapping experts, spanning multiple domains, including biodiversity, cultural heritage, and others.

For the purposes of our study, we focus exclusively on schema mappings related to the cultural heritage domain, specifically those using the ISO 21127:2023 CIDOC CRM ontology. This restriction narrows the scope of our collection but ensures domain consistency and ontological uniformity, allowing a meaningful evaluation. We consider this curated subset a *gold standard* for benchmarking LLM-generated schema mappings. More specifically, the mappings we use describe the transformation of diverse entity types such as persons, locations, organisations, and events. They were developed in the context of projects like RICONTRANS[3], and ARIADNEplus[4], and the China Biographical Database Project (CBDB)[5]. In total, our evaluation set includes 13 distinct schema mapping projects, comprising 53 mapping domains and 308 mapping links, in total.

In our setting, the generated output is considered successful only if it satisfies three complementary requirements: it must be syntactically valid with respect to the X3ML schema, semantically aligned with the target ontology, and structurally coherent as an executable mapping specification. Accordingly, the evaluation criteria below distinguish between syntactic validity, target-resource validity, correspondence correctness, ontological connectivity, and value generation validity.

Given a generated mapping project (M_{AI}) and its corresponding gold-standard mapping project (M_{GS}), the main criteria for assessing mapping quality are:

(1) **X3ML validity**. This criterion assesses whether M_{AI} conforms to the X3ML schema and can be parsed and executed by the X3ML Engine.
(2) **Target resource validity**. This criterion assesses whether the target classes and properties used in M_{AI} exist in the target ontology and are syntactically correct CIDOC CRM resources.
(3) **Valid mapping (class resources)**. This criterion assesses whether each target class generated in M_{AI} corresponds to the class used in M_{GS} for the same source node.
(4) **Valid mapping (property resources)**. This criterion assesses whether each target property generated in M_{AI} corresponds to the property used in M_{GS} for the same source relation or value-bearing element.
(5) **Target resources connectivity**. This criterion assesses whether the generated target path is structurally valid with respect to CIDOC CRM, i.e. whether the selected properties connect compatible domain and range classes.
(6) **Valid generation of values**. This criterion assesses whether the generator policy declarations in M_{AI} are executable and can produce valid URIs or literal values from source values or constants.

[3]https://ricontrans-project.eu/
[4]https://ariadne-infrastructure.eu/
[5]https://projects.iq.harvard.edu/cbdb/home

4.2 Quality Metrics

To evaluate the accuracy of the different methods, we introduce a set of quality metrics that quantify syntactic validity, exact correspondence, and partial correctness between the mappings found in the gold standard (M_{GS}) and the ones generated by an AI-based algorithmic method (M_{AI}).

To define the metrics formally, we first introduce a few notations of X3ML schema mappings. The basic construct is the notion of a *mapping*, which represents a semantic correspondence between elements of a source schema and a target schema. Each mapping is defined over what is called *domain*, which is used to map nodes between the two schemata, and is associated with a set of *links*, that are responsible for aligning the relations and their associated nodes.

Formally, we shall use S to denote a *schema*, where a schema consists of a set of *nodes*, denoted by $n(S)$, and a set of *relations* among the nodes, denoted by $r(S)$. In the context of a mapping project, let S be the source schema, and T the target schema. A mapping, from S to T, is actually a pair $M_{S,T} = (m_d, m_l)$ where m_d maps nodes of S to nodes of T, and m_l maps (relation, node)-pairs of S to (relation, node)-pairs of T. In particular, m_d is a partial function $m_d : n(S) \rightarrow n(T)$, while m_l is a partial function $m_l : r(S) \times n(S) \rightarrow r(T) \times n(T)$. Let $cont(M_{S,T})$ be the union of all the functions, i.e. $cont(M_{S,T}) = m_d \cup m_l$. We can now define the following metrics:

• **Syntactic Validity (SV)**. This metric examines the syntactic validity of the generated schema mappings. A mapping is considered valid if it conforms to the X3ML schema and can be processed by the X3ML engine without errors. It is defined as:

$$SV(M_{AI}) = \begin{cases} 1 & \text{if } \mathrm{M}_{AI} \text{ conforms to the X3ML schema (XSD)} \\ 0 & \text{otherwise} \end{cases}$$

• **ExactMatch (EM)**. This metric checks if two mapping projects are exactly equal, thus it is a binary metric defined as:

$$EM(M_{AI}, M_{GS}) = \begin{cases} 1 & \text{if } (\mathrm{cont}(\mathrm{M}_{AI}) == cont(M_{GS})) \\ 0 & \text{otherwise} \end{cases}$$

• **Accuracy (A).** It measures the proportion of correct mappings among all generated mappings, and it is defined as:
$A(M_{AI}, M_{GS}) = \frac{|cont(M_{AI}) \cap cont(M_{GS})|}{|cont(M_{AI})|}$.

Syntactic validity (SV) is the baseline measure that ensures that the generated schema mappings conform to the strict syntactic constraints of the X3ML language and can be processed by downstream data transformation pipelines (e.g. using *X3ML Engine*). It directly supports the assessment of X3ML validity and, to the extent that generator declarations affect execution, valid value generation. Since invalid mappings cannot be parsed and executed, SV constitutes a prerequisite for any further evaluation. ExactMatch (EM) is a strict binary metric requiring the generated mapping to be identical to the corresponding gold-standard mapping, thus capturing complete agreement across target resources, structural organization, and mapping components. In contrast, Accuracy (A) provides a more fine-grained assessment by measuring the proportion of correctly generated mapping elements that match those in the gold standard. It therefore captures partial correctness with respect to

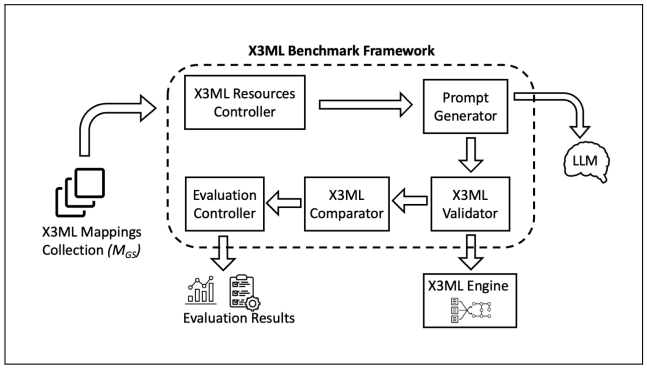

Figure 2: A general overview of the benchmark framework and the workflow

target-resource selection, class and property correspondences, and target-resource connectivity as represented in the generated mapping paths. Accuracy is defined at the mapping-component level rather than at the RDF triple level because the object of evaluation is the generated X3ML specification itself, not the RDF graph produced after the transformation execution. Triple-level comparison would also conflate mapping quality with source-data coverage, URI generation choices, and alternative but valid modelling patterns.

While additional metrics, such as recall and F1-score, could provide further insights, this work focuses on a precision-oriented evaluation to assess the correctness of generated mapping elements under strict syntactic and semantic constraints. A more extensive exploration of complementary metrics, including recall-oriented measures and alternative valid mappings, is left for future work.

4.3 X3MLBench Software

To assess the ability of different LLMs to generate accurate X3ML schema mappings, we developed a Python-based evaluation framework that systematically measures the quality of the generated mappings. Its primary purpose is to facilitate the comparison between automatically generated mappings and manually curated ones from the gold standard.

X3MLBench is depicted in Figure 2. The software starts by reading a structured directory containing gold standard mapping projects, each comprising source data and manually created X3ML schema mapping resources. It then invokes different LLMs to automatically generate mappings for the same data sources. Once the mappings are produced, the tool performs a detailed comparison between the generated and gold standard mappings. This comparison is carried out using the established evaluation metrics that were given in the previous section. In addition to computing these metrics, the tool generates detailed evaluation reports that include the actual values of the metrics as well as visual plots. These outputs allow users to visually interpret the performance of different models and configurations. The source code, the gold standard datasets and the detailed evaluation results can be found on GitHub[6].

Beyond supporting the experiments reported here, X3MLBench is intended as an extensible evaluation testbed for future schema mapping generation methods, including retrieval-augmented generation [15], fine-tuned smaller language models, and hybrid approaches that combine LLM generation with formal validation and feedback mechanisms.

5 Evaluation

5.1 Evaluated LLMs

For this study, we employed a diverse set of state-of-the-art LLMs to automatically generate X3ML schema mappings. Specifically, we used:

- OpenAI's GPT 5.3 Instant[7] [30] is a widely adopted family of models known for their strong general-purpose understanding and generation capabilities across diverse tasks;
- Gemini 3 Flash [8] [31] is a multimodal AI model, designed for frontier-level reasoning and agentic workflows at high speed and low cost.
- DeepSeek-V3-0324[9] [17] is a model optimised for code programming-related tasks, with a focus on code understanding, generation and technical reasoning;
- Mistral-Medium-2505[10] is a family of open-source and commercial LLMs recognized for their enhanced multilingual support, code generation and reasoning performance;
- Grok-3[11] is a conversational AI assistant developed by xAI, with emphasis on providing real time information.
- Llama-4-Scout-17B-16E-Instruct[12] [32] is a family of open-weight transformer-based LLMs designed for efficient training and performance across a range of natural language processing tasks.

This diversity of LLMs enables us to investigate how differences in model architectures, training objectives, and design philosophies impact the quality of automatically generated schema mappings. By including both proprietary and open-weight models, we aim to capture a broad range of capabilities and behaviors. For each mapping project, we use an identical prompt across all models, ensuring that the evaluation of the generated mappings is based on consistent input (e.g. the source data and the prompt). This controlled setup allows us to isolate the effect of the model itself on mapping quality. Moreover, the comparison of the outputs helps identify strengths and weaknesses particular to certain models, informing future model selection or prompt design in similar tasks.

In our experiments, we evaluate the impact of different prompting strategies on the quality of the generated schema mappings. All prompts provide the same sample input data in XML format, and instruct the LLM to produce schema mappings using X3ML as the mapping language, and CIDOC CRM as target ontology. The three prompt types vary in the amount and type of the guidance provided as follows:

- **Zero-Shot prompt** includes only the XML source data and a brief instruction to implement the schema mappings in X3ML targeting the CIDOC CRM ontology. No additional examples

[6] https://github.com/ymark/x3ml_comparator

[7] https://openai.com/index/gpt-5-3-instant/
[8] https://deepmind.google/models/gemini/flash/
[9] https://huggingface.co/deepseek-ai/DeepSeek-V3-0324
[10] https://docs.mistral.ai/getting-started/models/models_overview/
[11] https://x.ai/grok
[12] https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E-Instruct

or context is given. This prompting strategy is used to test the model's ability to infer the mapping structure and syntax.
- **Syntax-Aware prompt** includes a small indicative X3ML snippet as an example. The snippet is syntactically correct but irrelevant or limited in scope serving as guidance for understanding the structure and format of X3ML.
- **In-Context Mapping prompt** includes a complete domain-relevant mapping project alongside the XML source data. This prompt offers both syntactic and semantic guidance by illustrating real mapping definition in the context of similar source data. It is used to assess the model's ability to generalize schema mapping in a guided manner.

The actual prompts that were used for the evaluation for different tasks can be found at `X3MLBench` GitHub repository.

5.2 Lexical Baseline Approach

To provide a point of reference for evaluating the performance of LLM-based schema mapping generation, we implement a simple heuristic baseline based on lexical similarity. The baseline focuses on the mapping selection task, i.e. the identification of appropriate ontology classes and properties for elements of the source schema. Specifically, for each element in the input XML data, the baseline selects candidate classes and properties from the CIDOC CRM ontology by computing lexical similarity between the element name and the labels of the ontology resources. The resource with the highest similarity score is selected as the mapping target. This process is applied independently for each element, without considering structural relationships or contextual information. This approach is inspired by the traditional schema matching techniques based on lexical similarity [29].

For the implementation of the baseline, we developed a lightweight Python script that extracts the element names from the XML input, normalizes them (e.g., tokenization and lowercasing), and matches them against CIDOC CRM classes and properties using lexical similarity. For each element, the most similar class and property are selected and compared against the gold-standard mappings. Lexical similarity is computed using normalized string similarity between the XML element name and the ontology resource label. We use this simple baseline intentionally as a lower-bound reference for the mapping selection subtask, since more sophisticated ontology matching systems typically produce correspondences between schema entities but do not directly address the complete X3ML generation task considered here.

It should be noted that the baseline does not guarantee complete X3ML mapping files and therefore does not guarantee syntactic validity with respect to the X3ML schema. As a result, it is evaluated only with respect to mapping accuracy, and not on syntactic validity. This distinction highlights the added complexity of LLM-based approaches, which must simultaneously address both semantic alignment and structured output generation. While simplistic, this baseline captures a purely lexical mapping strategy based solely on surface-level lexical resources. As such, it provides a lower bound for performance, allowing us to assess whether LLMs can go beyond straightforward name-based matching when generating schema mappings.

5.3 Evaluation Results

Before analyzing mapping accuracy, we first examine the syntactic validity of the generated X3ML mappings. This is a prerequisite for usability, as invalid mappings cannot be parsed and executed by the X3ML engine. We evaluate this aspect using a Syntactic Validity (SV) rate, which measures the proportion of generated mappings that conform to the X3ML schema. We observe that the validity rates vary significantly across prompting strategies. Zero-shot prompting results in a high proportion of invalid mappings, as the models fail to adhere to the required X3ML structure. In contrast, syntax-aware and in-context prompting substantially improve validity, indicating that explicit structural guidance is essential for generating well-formed outputs. Since invalid mappings are treated as incorrect, syntactic validity directly impacts the overall accuracy of the generated mappings.

To evaluate the effectiveness of our prompting strategies, we conducted a quantitative analysis based on the quality metrics. Specifically, we conducted experiments using the three different prompting strategies across the evaluated LLMs, utilizing resources from the gold standard (M_{GS}). In addition, we compare the LLM-based approaches against a lexical baseline, which relies solely on surface-level string similarity for mapping selection and does not consider structural or contextual information.

We computed the average accuracy of each prompting method by aggregating the accuracy scores obtained per LLM and dataset. The results are depicted in Figure 3. As expected, Zero-Shot prompt achieved the lowest accuracy (i.e. very close to zero) primarily due to the lack of contextual information about the X3ML syntax. In many cases, this resulted in invalid or incorrectly structured X3ML schema mappings. The Syntax-Aware prompt, which includes sample X3ML mappings, improved the models' ability to generate syntactically valid outputs and thus increased the overall accuracy. Finally, the In-Context Mapping prompt slightly improved the average accuracy score, mainly due to the provision of contextual information in the form of a relevant X3ML mapping file. This additional context enabled the LLMs to better understand the domain, identify key concepts and generate more accurate schema mappings.

Compared to the lexical baseline, which relies solely on name-based matching, LLM-based approaches achieve higher accuracy across most datasets (see Table 1). While the lexical baseline approach performs reasonably well on datasets with descriptive element names, its performance degrades significantly in more complex or ambiguous cases that require contextual interpretation. This highlights the limitations of purely lexical matching approaches. On average, the lexical baseline achieves substantially lower accuracy than all LLM-based approaches, with scores ranging from 0 to 0.36 and an average of 0.14. This reinforces its role as a lower-bound reference for this benchmark. In contrast, LLM-based approaches are able to leverage contextual and structural information, enabling them to capture semantic relationships beyond surface-level similarity and produce more accurate mappings. Notably, the lexical baseline does not generate full X3ML mappings and therefore is evaluated only with respect to mapping accuracy, whereas LLM-based approaches address both mapping selection and X3ML syntax compliance.

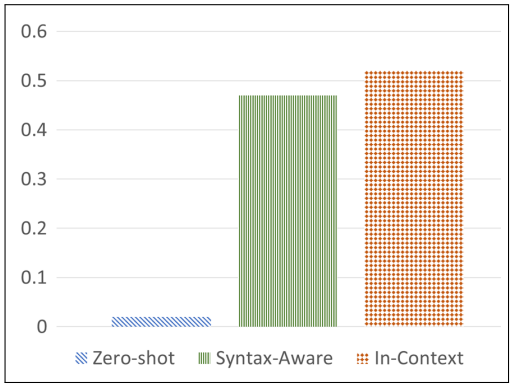

Figure 3: Average accuracy score using different prompting methods

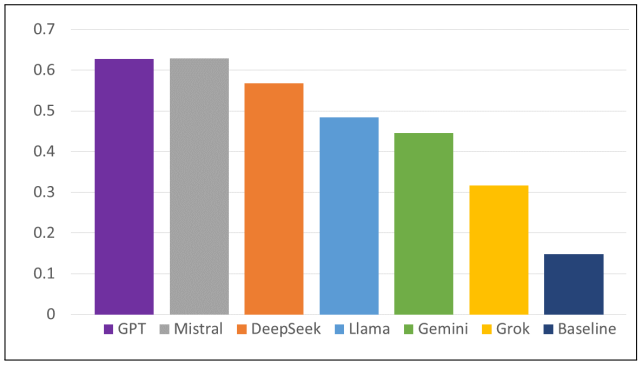

Figure 4: Average accuracy score for the evaluated LLMs (using in-context prompt) and the lexical baseline approach

Table 1 presents the accuracy scores achieved by each model and the lexical baseline across the evaluated datasets. The highest-scoring model for each dataset is highlighted in bold. From that table and the aggregated values per LLM depicted in Figure 4, we observe that GPT 5.3 and Mistral outperform others, followed by DeepSeek-V3, and Llama. The one with the lowest accuracy score is Grok-3.

In the following, we present detailed accuracy results for each LLM and dataset from M_{GS}. These results are visualized in Figure 5. We observe that, for certain datasets, all LLMs achieved consistently high accuracy scores. This can be attributed to the characteristics of the datasets themselves; the corresponding XML files contained element names that were sufficiently descriptive and recognizable, allowing the LLMs to correctly interpret their semantics and generate accurate schema mappings aligned with the CIDOC CRM ontology.

For instance Dataset #3, contains XML elements titled `Person` providing a clear indication of the type of resource and the corresponding target class in CIDOC CRM (i.e. `E21_Person`). Furthermore, the child elements are similarly descriptive, including element names such as `IdentificationNumber`, suggesting a unique identifier for the person, `Appellation` indicating the person's name,

	Baseline	GPT	Gemini	DeepSeek	Mistral	Grok	Llama
#1	0.0	**0.45**	0.33	**0.45**	0.4	0.39	0.35
#2	0.27	0.58	0.33	0.4	**0.6**	0.0	0.52
#3	0.26	**0.9**	0.5	**0.9**	0.8	0.8	0.73
#4	0.12	**0.7**	0.55	0.12	0.57	0.31	0.07
#5	0.07	0.4	0.29	0.79	**0.86**	0.0	0.0
#6	0.23	**0.9**	0.6	0.82	0.85	0.0	0.74
#7	0.36	**0.94**	0.38	0.71	0.71	0.8	0.71
#8	0.22	**0.9**	0.7	0.65	0.8	0.65	0.71
#9	0.15	0.8	0.67	**0.93**	0.8	0.6	0.79
#10	0.15	0.61	0.55	0.59	0.73	0.0	**0.95**
#11	0.0	0.3	0.23	0.36	**0.5**	0.25	0.27
#12	0.08	0.53	**0.57**	**0.57**	0.43	0.24	0.36
#13	0.0	**0.15**	0.1	0.09	0.12	0.08	0.1
avg	0.14	**0.63**	0.45	0.57	**0.63**	0.32	0.48

Table 1: Accuracy scores using the In-Context Mapping prompt and the lexical baseline across all datasets

`Type` denoting a categorization, and `Description`. These semantically rich labels help guide the LLMs toward identifying the proper classes from the target ontology and eventually generating relevant schema mappings. A similar pattern is observed in datasets #7, #8, #9, and #10 which include elements with names `Person`, `Location`, `Collection`, and `Event` respectively, each directly aligning with well-defined CIDOC CRM classes. Also in several cases the actual values (i.e. text nodes) of those elements were also descriptive enough.

In contrast, dataset #13 presents a more challenging case due to its lack of descriptive element names, which forced the LLMs to make more arbitrary assumptions. This dataset contains information about ancient Chinese kingdoms, yet the core elements are labelled generically as `KIN_DATA`, offering little semantic clarity. Without additional contextual or descriptive information, it becomes significantly more difficult for the models to infer the intended meaning of the elements and produce accurate mappings. Nevertheless, some of the evaluated LLMs were still able to generate partially meaningful schema mappings. In these cases, the lexical baseline also performs poorly, further confirming that simple name-based matching is insufficient when semantic cues are weak or ambiguous.

It is worth noting that there are a few experiments that failed to generate meaningful mappings; in these cases, the accuracy score is 0. The reason for failure in all these cases is the construction of an X3ML mapping file that is structurally not valid with respect to the definition of the X3ML mapping language. We consider those cases as invalid, considering that they cannot be directly used for transforming data using X3ML Engine, towards automating the data transformation process.

To better understand model behavior, we conducted a qualitative analysis of common error patterns observed in the generated schema mappings. We found that errors frequently arise from: (i) incorrect selection of ontology classes or properties, particularly when source element names are ambiguous; (ii) improper structural connections between target entities, leading to invalid or semantically inconsistent mappings; and (iii) missing or inconsistent generator policy references. In addition, some models produce syntactically valid mappings that violate ontological constraints. These observations indicate that the errors are not entirely random,

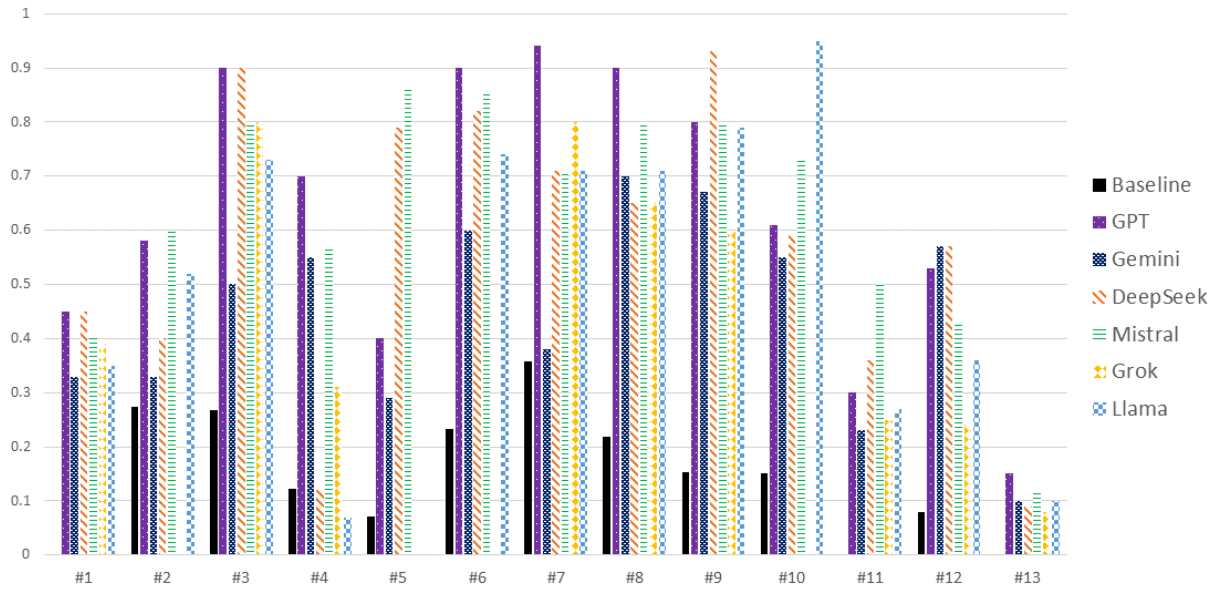

Figure 5: Accuracy per dataset for the evaluated LLMs (using in-context prompt) and the lexical baseline

but are influenced by source-schema clarity, prompt richness, and the structural complexity of the target CIDOC CRM pattern.

6 Concluding Remarks

In this paper, we investigated the extent to which LLMs can support the generation of X3ML schema mappings that specify correspondences between structured source schemata and ontology-based target representations. We formulated schema mapping generation as a structured output task under strict syntactic and semantic constraints, and introduced a curated benchmark of real-world schema mapping projects expressed in the X3ML language to enable systematic evaluation. Using this benchmark, we evaluated multiple state-of-the-art LLMs under different prompting strategies, providing an empirical assessment of their ability to generate valid and semantically meaningful schema mappings. The empirical evaluation conducted using the `X3MLBench` reveals several key insights into the performance of LLMs for schema mapping generation:

(1) **Impact of Prompting Strategies**: Prompt design was the most critical factor for success. Zero-shot prompting proved ineffective, yielding accuracy scores near zero due to a lack of structural context. Syntax-aware prompting significantly improved the models' ability to generate well-formed X3ML code, while In-Context Mapping (few-shot) achieved the highest performance by providing both syntactic and semantic guidance.

(2) **LLM Performance Comparison:** Under optimal in-context prompting, GPT 5.3 Instant and Mistral-Medium-2505 emerged as the top performers, both achieving an average accuracy of 0.63. They were followed by DeepSeek-V3 (0.57), Llama-4-Scout (0.48), and Gemini 3 Flash (0.45). Grok-3 demonstrated the lowest performance with an average accuracy of 0.32.

(3) **Superiority Over Lexical Baselines:** All evaluated LLMs consistently outperformed the lexical similarity baseline, which averaged only 0.14 accuracy. This confirms that LLMs are capable of capturing complex semantic relationships and structural constraints that surface-level name matching cannot address.

(4) **Semantic Clarity and Data Quality:** Accuracy was highly dependent on the descriptive quality of the source data. Datasets with semantically rich labels (e.g., "Person," "Location") saw accuracy scores as high as 0.95 (e.g., Llama on Dataset 10), whereas datasets with opaque labels (e.g., "KIN_DATA" in Dataset 13) resulted in significantly lower scores across all models (ranging from 0.08 to 0.15).

(5) **Syntactic and Semantic Integrity:** While LLMs achieved an overall average accuracy of 0.53 with in-context prompting, failures typically occurred due to the generation of invalid X3ML structures or violations of ontological constraints (improper class/property connectivity).

These results provide broader insights into the behavior of LLMs in structured semantic generation tasks. In particular, they indicate that, while LLMs can assist in producing formal representations, their effectiveness depends heavily on explicit structural patterns and contextual grounding. This suggests that fully end-to-end approaches may be insufficient for reliable data integration, and supports the use of intermediate, interpretable representations (i.e., schema mappings), that enable validation, control, and reproducibility within the transformation pipeline.

The experimental findings suggest that because LLMs can produce a substantial portion of mapping specifications automatically, the primary role of domain experts will shift from construction to the refinement and validation of initial drafts. This transition to semi-automated workflows has the potential to significantly reduce the labor-intensive effort and high costs traditionally associated with semantic data integration while maintaining the necessary reliability for real-world applications. Although our experiments

focus on CIDOC CRM, X3ML has also been used with other domain ontologies, such as MarineTLO [33], MINGEI ontology [36], SeaLiT ontology [7], AO-Cat ontology [8]. This suggests that our work may be transferable to other ontology-based mapping scenarios, provided that sufficient contextual information about the target mapping language and ontology is included in the prompt.

As future work, we plan to investigate iterative self-verification and reflection-based prompting, where generated mappings are checked and revised against criteria such as completeness, ontological connectivity, and semantic precision. We also plan to explore fine-tuning, constrained decoding [10], or hybrid symbolic-neural methods [13]. Moreover, future versions could distinguish accuracy at different granularity levels, for example by reporting separate scores for correctly generated target classes and properties.

Acknowledgments: This work was supported by the ECHOES project, funded by the European Union under the Horizon Europe programme, Grant Agreement No. 101157364.

References

[1] Zohra Bellahsene, Angela Bonifati, Fabien Duchateau, and Yannis Velegrakis. *On evaluating schema matching and mapping*. Springer, 2011.

[2] Christopher Buss, Mahdis Safari, Arash Termehchy, David Maier, and Stefan Lee. Towards scalable schema mapping using large language models. In *Proceedings of the 1st workshop connecting academia and industry on Modern Integrated Database and AI Systems*, pages 12–15, 2025.

[3] Salvatore Carta, Alessandro Giuliani, Leonardo Piano, Alessandro Sebastian Podda, Livio Pompianu, and Sandro Gabriele Tiddia. Iterative zero-shot llm prompting for knowledge graph construction. *arXiv preprint arXiv:2307.01128*, 2023.

[4] Michelle Cheatham and Catia Pesquita. Semantic data integration. *Handbook of big data technologies*, pages 263–305, 2017.

[5] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. Rml: A generic language for integrated rdf mappings of heterogeneous data. *Ldow*, 1184, 2014.

[6] Martin Doerr. The cidoc conceptual reference module: an ontological approach to semantic interoperability of metadata. *AI magazine*, 24(3):75–75, 2003.

[7] Pavlos Fafalios, Athina Kritsotaki, and Martin Doerr. The sealit ontology–an extension of cidoc-crm for the modeling and integration of maritime history information. *ACM Journal on Computing and Cultural Heritage*, 16(3):1–21, 2023.

[8] Achille Felicetti, Carlo Meghini, JD Richards, and Maria Theodoridou. The ao-cat ontology. *Zenodo*, 2023.

[9] Johannes Frey, Lars-Peter Meyer, Natanael Arndt, Felix Brei, and Kirill Bulert. Benchmarking the abilities of large language models for rdf knowledge graph creation and comprehension: How well do llms speak turtle? *arXiv preprint arXiv:2309.17122*, 2023.

[10] Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952, 2023.

[11] Skander Ghazzai, Daniela Grigori, Boualem Benatallah, and Raja Rebai. Harnessing gpt for data transformation tasks. In *2024 IEEE International Conference on Web Services (ICWS)*, pages 1329–1334. IEEE, 2024.

[12] Pieter Heyvaert, Ben De Meester, Anastasia Dimou, and Ruben 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*, pages 213–217. Springer, 2018.

[13] P Hitzler, MK Sarker, TR Besold, AD Garcez, S Bader, H Bowman, P Domingos, P Hitzler, KU Kühnberger, LC Lamb, et al. Neural-symbolic learning and reasoning: A survey and interpretation. *Frontiers in artificial intelligence and applications*, 342:1–51, 2022.

[14] Marvin Hofer, Johannes Frey, and Erhard 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.

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[16] Huahang Li, Longyu Feng, Shuangyin Li, Fei Hao, Chen Jason Zhang, and Yuanfeng Song. On leveraging large language models for enhancing entity resolution: a cost-efficient approach. *arXiv preprint arXiv:2401.03426*, 2024.

[17] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[18] Frank Manola, Eric Miller, Brian McBride, et al. Rdf primer. *W3C recommendation*, 10(1-107):6, 2004.

[19] Yannis Marketakis, Nikos Minadakis, Haridimos Kondylakis, Konstantina Konsolaki, Georgios Samaritakis, Maria Theodoridou, Giorgos Flouris, and Martin Doerr. X3ml mapping framework for information integration in cultural heritage and beyond. *International Journal on Digital Libraries*, 18:301–319, 2017.

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

[21] Apostolos Mavridis, Stergios Tegos, Christos Anastasiou, Maria Papoutsoglou, and Georgios Meditskos. Large language models for intelligent rdf knowledge graph construction: results from medical ontology mapping. *Frontiers in Artificial Intelligence*, 8:1546179, 2025.

[22] Michalis Mountantonakis and Yannis Tzitzikas. Large-scale semantic integration of linked data: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–40, 2019.

[23] Anmol Nayak and Hari Prasad Timmapathini. Llm2kb: Constructing knowledge bases using instruction tuned context aware large language models. *arXiv preprint arXiv:2308.13207*, 2023.

[24] Sanaz Saki Norouzi, Adrita Barua, Antrea Christou, Nikita Gautam, Andrew Eells, Pascal Hitzler, and Cogan Shimizu. Ontology population using llms. *arXiv preprint arXiv:2411.01612*, 2024.

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

[26] Vetle Ryen, Ahmet Soylu, and Dumitru Roman. Building semantic knowledge graphs from (semi-) structured data: a review. *Future Internet*, 14(5):129, 2022.

[27] Wilma Johanna Schmidt, Irlan Grangel-González, Tobias Huschle, Lena Wagner, Evgeny Kharlamov, and Adrian Paschke. Llm-supported mapping generation for semantic manufacturing treasure hunting. In *European Semantic Web Conference*, pages 84–101. Springer, 2025.

[28] Ankita Sharma, Xuanmao Li, Hong Guan, Guoxin Sun, Liang Zhang, Lanjun Wang, Kesheng Wu, Lei Cao, Erkang Zhu, Alexander Sim, et al. Automatic data transformation using large language model-an experimental study on building energy data. In *2023 IEEE International Conference on Big Data (BigData)*, pages 1824–1834. IEEE, 2023.

[29] Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. In *Journal on data semantics IV*, pages 146–171. Springer, 2005.

[30] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[31] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[33] Yannis Tzitzikas, Carlo Allocca, Chryssoula Bekiari, Yannis Marketakis, Pavlos Fafalios, Martin Doerr, Nikos Minadakis, Theodore Patkos, and Leonardo Candela. Integrating heterogeneous and distributed information about marine species through a top level ontology. In *Research conference on metadata and semantic research*, 2013.

[34] Dylan Van Assche, Thomas Delva, Gerald Haesendonck, Pieter Heyvaert, Ben De Meester, and Anastasia Dimou. Declarative rdf graph generation from heterogeneous (semi-) structured data: A systematic literature review. *Journal of Web Semantics*, 75:100753, 2023.

[35] Guohui Xiao, Lin Ren, Guilin Qi, Haohan Xue, Marco Di Panfilo, and Davide Lanti. Llm4vkg: Leveraging large language models for virtual knowledge graph construction. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI)*, 2025.

[36] Xenophon Zabulis, Nikolaos Partarakis, Carlo Meghini, Arnaud Dubois, Sotiris Manitsaris, Hansgeorg Hauser, Nadia Magnenat Thalmann, Chris Ringas, Lucia Panesse, Nedjma Cadi, et al. A representation protocol for traditional crafts. *Heritage*, 5(2):716–741, 2022.

[37] Runhao Zhao, Weixin Zeng, Jiuyang Tang, Yawen Li, Guanhua Ye, Junping Du, and Xiang Zhao. Towards unsupervised entity alignment for highly heterogeneous knowledge graphs. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pages 3792–3806. IEEE Computer Society, 2025.

[38] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[39] Yikai Zheng, Wanquan Liu, Bi Zeng, Yichun Feng, Xiawei Du, Lu Zhou, and Yixue Li. Automating the construction of contextualized biomedical knowledge graphs for scientific inference. *bioRxiv*, pages 2026–01, 2026.