

Beyond Prompt-to-RDF: A Vision for Scalable and Explainable Graph Transformations via LLM-assisted Schema Mappings

Yannis Marketakis^{1,2,*}, Yannis Tzitzikas^{1,2}

¹*Institute of Computer Science, Foundation for Research and Technology – Hellas, Heraklion, Greece*

²*Computer Science Department, University of Crete, Heraklion, Greece*

Abstract

The increasing adoption of Large Language Models (LLMs) has led to a growing number of prompt-to-RDF approaches that rely on LLMs to directly transform heterogeneous data collections into RDF graph representations. While effective for rapid prototyping, such approaches raise concerns regarding transparency, reproducibility, and maintainability, as modeling decisions remain implicit and difficult to validate or reuse. This vision paper argues for repositioning LLMs in graph transformation workflows, from black-box data transformers to assistive generators of schema mappings. In the proposed approach, LLMs are used to produce schema mappings, a task that is traditionally manual and labor-intensive, while data transformation is delegated to established data transformation frameworks. By doing so, the schema mapping process can be significantly accelerated, addressing key bottlenecks in large-scale semantic integration pipelines. The paper grounds this vision in ongoing work that explores the use of different LLMs and prompting strategies to generate schema mappings using X3ML mapping framework. While a full experimental evaluation is beyond the scope of this paper, this work illustrates the feasibility of the approach. Overall, the paper outlines a research agenda towards more explainable, trustworthy and semantically grounded LLM-assisted transformation pipelines.

Keywords

Large Language Models, Schema Mapping Generation, RDF Graph Transformation

1. Introduction

The transformation of heterogeneous data into RDF graph representations plays a central role in semantic data integration, knowledge graph construction and interoperability across information systems. By lifting data from relational databases, tabular files, APIs, and semi-structured formats into RDF graphs aligned with shared ontologies, organizations can enable semantic querying, reasoning, data re-use, and cross-domain integration. As a result, the transformation of data to RDF-based graphs has become a foundational component of data infrastructures across various domains such as cultural heritage, life sciences, e-government and others.

At the same time, the scale and diversity of data landscapes across different disciplines, pose significant challenges for RDF graph construction. Large volumes of heterogeneous data are continuously produced, updated, and exchanged. In this context, through the recent advances in Large Language Models (LLMs), emerged a new alternative for constructing RDF knowledge graphs. These approaches have explored the use of LLMs as tools to directly transform raw data into RDF graphs. While this approach is quick and attractive for small-scale scenarios and rapid prototyping, it has serious limitations over massive and continuously evolving datasets. Furthermore, it is computationally expensive and difficult to sustain in production systems. In addition, repeatedly invoking LLMs for data transformations raises concerns regarding cost, latency and operational scalability. Similar concerns have been raised in recent work on deep learning, which highlights the need for scalable graph-based abstractions over relational data, rather than relying on costly instance-level processing pipelines [1].

Published in the Proceedings of the Workshops of the EDBT/ICDT 2026 Joint Conference (March 24-27, 2026), Tampere, Finland

*Corresponding author.

✉ marketak@ics.forth.gr (Y. Marketakis); tzitzik@ics.forth.gr (Y. Tzitzikas)

ORCID 0000-0002-0417-2526 (Y. Marketakis); 0000-0001-8847-2130 (Y. Tzitzikas)



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

In contrast, schema mappings provide a well-established mechanism for specifying how source data structures correspond to target schemata (i.e. ontologies). Such mappings make modeling decisions explicit, enable the reuse across datasets, and support validation, adaptation, debugging, and maintenance of transformation pipelines. After the definition of schema mappings, dedicated data transformation software components, use them to transform data and construct RDF knowledge graphs. However, the definition of schema mappings remains a manual and expertise-intensive process, often constituting a major bottleneck in semantic integration workflows. More specifically, the definition of schema mappings requires expertise and knowledge of: (a) the domain and the schema of the source data, (b) the target ontology, and (c) the schema mapping technology. Given the volume and heterogeneity of today’s data, the manual generation of schema mappings does not scale.

This paper argues that automating the construction of schema mappings is a more effective and sustainable strategy than applying LLMs directly to data transformation. By leveraging LLMs to assist in generating schema mappings, rather than transforming data themselves, semantic data integration pipelines can combine scalability and efficiency with transparency and semantic validation. This vision positions LLMs as accelerators of human modeling effort, enabling the definition of explicit, reusable schema mappings that can be exploited by existing transformation frameworks and tools over large volumes of data.

The contributions of our work are: (a) a conceptual reframing of the role of LLMs for graph transformations, positioning themselves as assistants for schema mapping generation, rather than data transformation engines; (b) an analysis of the limitations of prompt-to-RDF approaches; and (c) a grounded vision for scalable and reusable semantic data integration based on X3ML schema mapping framework.

The paper is organized as follows: Section 1 introduces the objectives of the paper and outlines its main contributions. Section 2 provides background information and details about related works. Section 3 elaborates on the limitations of prompt-to-RDF approaches. Section 4 presents our vision of repositioning LLMs as schema mapping accelerators. Section 5 grounds this vision by discussing our experience to date with an existing schema mapping framework and LLMs. Finally, Section 6 concludes and identifies directions for further research.

2. Background & Related work

The construction of RDF knowledge graphs through data transformation from heterogeneous data sources has been extensively studied in the literature (e.g. the survey in [2]). A dominant approach relies on the use of schema mapping languages, which declaratively specify how data structured according to a source schema are transformed into resources conforming to a target schema or ontology. Such languages define transformation rules in a structured, unambiguous manner, making modelling decisions explicit and machine-interpretable. Representative examples include X3ML language [3], RML [4], and R2RML [5]. These languages are supported by tools that execute the mappings to systematically produce RDF knowledge graphs.

More recently, several approaches have explored the use of LLMs to directly transform data into RDF. LLM2KB [6] proposes a system for constructing knowledge bases by relying on LLMs to generate RDF content, while other works, such as [7], adopt iterative refinement strategies to improve the generated outputs. [8] evaluates the ability of different LLMs in generating knowledge graphs serialized in Turtle syntax. SQLMorpher [9] focuses on automating data transformation from relational databases in the energy domain, whereas [10] studies the effectiveness of different GPT models across common data transformation tasks. Finally, [11] investigates the suitability of LLMs for populating knowledge graphs from structured data sources.

While both schema mapping-based approaches and LLM-based approaches aim to transform heterogeneous data into RDF knowledge graphs they differ in how transformation logic is expressed and applied. The implications of these differences are examined in the following section.

3. Limitations of Prompt-to-RDF Approaches

A key limitation of direct LLM-based data transformation is the absence of an explicit intermediate representation, that describe class assignments, property selections and others. In prompt-to-RDF workflows, such modeling decisions are embedded implicitly in the generated RDF triples. This lack of an explicit schema mapping layer makes it difficult to verify, review, or adjust transformation logic before its execution. As a result, validation typically occurs only after RDF graphs are constructed, at which point identifying and correcting potential errors becomes costly and impractical.

Another issue, closely related to the previous one, is the limited support for debugging and maintenance. When errors occur in RDF graphs produced by LLMs, tracing their origin is cumbersome, as there is no inspectable step that documents how source elements were transformed. In contrast to traditional schema mapping approaches, prompt-to-RDF transformations operate as black boxes, complicating error diagnosis and refinement of the transformation logic.

Another concern is variability in transformation results. LLM-based data transformations may yield different RDF graphs across runs, prompts, or model versions, even when applied to the same source data. Such variability undermines reproducibility, raises concerns for the accuracy of the generated graphs, and poses significant challenges for maintaining consistent RDF graph representations especially when applied over continuously evolving data sources.

Scalability and efficiency further limit the applicability of prompt-to-RDF approaches. Applying LLMs directly to large data collections requires repeated model invocations over large volumes of data, leading to high computational cost and latency. This is a problem that does not occur when source data are used for generating schema mappings, since in this case only a small subset or sample of the data is needed. Of course, even in this case, the transformation of large volumes of data has a computational cost, which is however lower and can scale using dedicated transformation frameworks, compared to LLMs.

Overall, these limitations highlight that directly relying on LLMs for data transformations mixes transformation logic with transformation execution. This coupling hinders verification, debugging, reproducibility, and scalability, motivating the need for alternative approaches that preserve explicit transformation logic while still get all the potential from LLM capabilities.

4. Vision: LLM-assisted Schema Mapping Generation

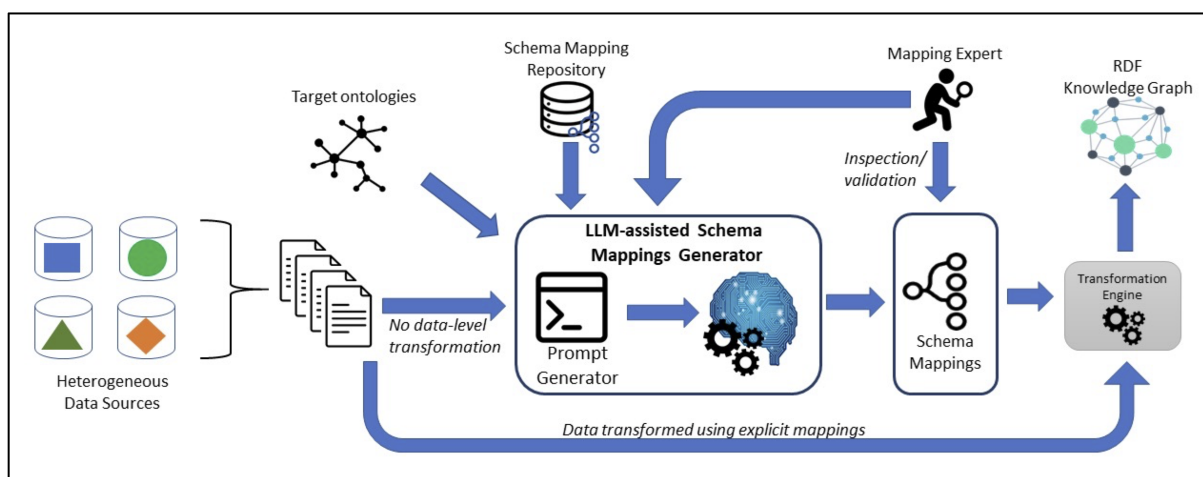


Figure 1: LLM-assisted generation of schema mappings, with iterative expert feedback, decoupling mapping definition from data-level transformation.

The aforementioned limitations indicate that the challenges of LLM-based data transformations stem

not from the actual use of LLM themselves, but from their direct application to data-level transformation. In particular, the absence of an explicit and inspectable transformation layer hinders validation, debugging and reuse of the transformation logic. This observation suggests that the strengths of LLMs (i.e. their ability to interpret the semantics of heterogeneous data) could be more effectively leveraged at the level of schema mapping generation rather than the data transformation execution. For this reason, we outline a vision for repositioning LLMs as assistants for generating explicit schema mappings, enabling scalable and explainable graph transformation pipelines while preserving the benefits of existing transformation frameworks.

Figure 1 illustrates the proposed vision; heterogeneous data sources are first collected and structurally normalized so that they can be consistently interpreted by the transformation engine used at execution time. Representative samples are selected to capture the structural characteristics and semantics of the source schema, as the objective is to derive mapping rules and correspondences, not to transform the entire dataset during prompting. A small subset or sample of those data, together with the target ontologies are then given as input to an LLM-assisted schema mapping generator component through a prompt construction process. Rather than operating on a full data collection, LLMs are guided using samples to generate explicit schema mappings that capture how source data elements correspond to classes, properties and relationships in the target ontologies. Crucially, no data-level transformation is performed by the LLM; the outcome of this process is the inspectable schema mappings, where expert validation results can be fed back to the generator to guide subsequent mapping generation rounds.

The schema mapping repository plays a central role in the proposed architecture by supporting and guiding the LLM-assisted schema mapping generation process. More specifically, the repository acts as a source of contextual knowledge that can be exploited to improve the accuracy and consistency of the generated mappings. Existing schema mappings can be retrieved based on their relevance with new data sources; for example, by comparing the structure or the contents of the source data, with the input data used in existing schema mappings (e.g. using text embeddings [12]). Then the selected ones can be provided as additional contextual input to the prompt construction process, enabling the LLM to reuse valid and assessed schema mappings. This approach facilitates the improvement, while preserving the explicit and inspectable nature of the generated schema mappings.

The generated schema mappings can be therefore reviewed, validated and refined by mapping experts prior to their execution, enabling early detection of modeling errors. Importantly, expert feedback resulting from this validation process can be fed back to the LLM-assisted schema mapping generation component, allowing the revision of the mappings based on curated human advice. Expert feedback is incorporated by adding corrected mappings fragments to subsequent prompts enabling iterative refinement of the generated schema mappings.

Upon the definition of the schema mappings and their revision/validation by mapping experts, the generated schema mappings can be used for the actual data transformation part. This is carried out without relying on the LLM, by delegating it to well-known data transformation engines (e.g. X3ML Engine [13] or RMLMapper [14]), that consume the validated schema mappings and apply them over large volumes of data to produce RDF graphs. Furthermore, this process can be repeated as needed, for example when new or updated data become available, without requiring invoking the LLM and regenerating the schema mappings.

This explicit feedback loop, together with the separation of mapping definition from transformation execution enables a scalable and explainable approach to graph transformation in which human expertise and LLM capabilities are effectively combined. This solution significantly accelerates the traditionally manual and time-consuming schema mapping definition process. At the same time, delegating the actual data transformations to dedicated transformation engines ensures efficiency, reproducibility, and robustness when operating over large and evolving data collections. Table 1 shows a side-by-side comparison with respect to different aspects.

Aspect	Prompt-to-RDF	LLM-Assisted Schema Mappings
Input	Full data	Sample data
Transparency	Implicit	Explicit
Reproducibility	Limited	High
Debugging	Limited	High
Variability	High	Limited
Scalability	Limited	High
Human control	Minimal	High

Table 1
Prompt-to-RDF versus LLM-assisted schema mapping generation

5. Grounding the Vision with X3ML Mapping Framework

To ground the proposed vision in practice, this section summarizes ongoing work that explores the use of LLMs for generating schema mappings based on X3ML framework [3]. X3ML provides a declarative mapping language and a transformation engine, and has been widely adopted in domains such as cultural heritage and biodiversity. Despite this support, defining schema mappings remains a manual and time-consuming task requiring expertise in both source schemata and target ontologies.

Figure 2 illustrates a minimal example of the proposed approach. The upper part consists of an XML source record describing a person with an identifier and a name. Based on this and the target ontology (e.g., CIDOC CRM [15] in this case), the LLM is prompted to generate an X3ML mapping that associates record elements with the class `E21_Person`, and then map the `id` element to an instance of `E42_Identifier` via the property `P1_is_identified`. The generated mapping constituted an explicit transformation specification that can be validated by mapping experts before being executed by the X3ML Engine for transforming all the conforming records into RDF triples. This example highlights the separation between mapping generation and data transformation execution, which enables transparency, validation and reuse at large scale.

In [16] we investigate whether LLMs can effectively assist in accelerating the definition of X3ML schema mapping by experimenting with multiple LLMs and prompting strategies. Five different LLMs, GPT4.1, DeepSeek-V3, Mistral, Grok-3, and Llama-4, were considered under three prompting techniques that reflect different levels of guidance. The zero-shot technique provides the LLM only with descriptions of the source data and the target ontology, assessing its ability to generate mappings without additional context. The syntax-aware technique augments the prompt with guidance on the structure of the X3ML mapping language, aiming to improve syntactic correctness and alignment with the expected mapping structure and format. Finally, in-context technique supplies the prompt with existing X3ML mappings that were defined for semantically and structurally similar source data, alongside with the current source data, enabling the LLM to reuse established modelling patterns.

In [16], schema mapping generation was assessed by comparing generated X3ML mappings against manually curated reference mappings. A mapping was considered correct if its generated mapping resources (e.g., source schema elements and target ontology resources) matched the reference mappings, and conformed to valid X3ML syntax. Accuracy therefore reflects the proportion of correctly generated mapping components in the reference mappings. This high-level metric is intended to provide indicative trends rather than a fine-grained performance comparison. To support transparency and reproducibility, the benchmark, the reference mappings, the prompts, and evaluation scripts are publicly available in an online repository¹.

Figure 3 provides an aggregated overview of the impact of the prompting techniques on the accuracy of the generated schema mappings. The figure is intended to illustrate general trends, rather than to support a detailed comparative evaluation that goes beyond the scope of this vision paper. As shown, zero-shot prompting yields the lowest accuracy, as it frequently generates schema mappings with invalid or incomplete structure with respect to X3ML schema. Syntax-aware prompting significantly improves the validity and usefulness of the generated mappings, by constraining the LLM output to the expected mapping format. In-context prompting technique further improves accuracy by guiding the LLM with representative examples, leading to schema mappings that align to existing modelling practices and consists of semantically correct components.

Our observations directly support the design choices outlined in Section 4. In particular the effectiveness of in-context prompting reinforces the central role of reusing existing schema mappings in guiding LLM-assisted mapping generation. Direct quantitative comparison with existing techniques is currently not possible, as there are no established baselines for this. Consequently, the reported accuracy values should be interpreted as indicative evidence of feasibility rather than competitive performance results. The observed average accuracy of 0.53 for the in-context strategy should be interpreted in relation to the complexity of schema mapping definition, which typically requires expert knowledge of the source domain and schemata, target ontologies and mapping languages. Achieving correct generation for more than half of mapping components without manual authoring constitutes a strong indication that the traditionally manual and time-consuming schema mapping process can be substantially accelerated, and in certain cases partially automated. We should however mention that this level of accuracy is not intended to replace expert involvement, but to provide high-quality initial mappings and reduce manual effort.

6. Conclusion and Research Agenda

This paper presented a vision for repositioning LLMs in graph transformation workflows, motivated by the scale and diversity of data landscapes. Given the massive volumes of heterogeneous data that must be integrated today, relying on LLMs to directly transform data into RDF knowledge graphs is impractical and highly resource-consuming. In this context, accelerating the traditionally manual and expertise-intensive task of schema mapping definition emerges as a sustainable strategy for enabling large-scale semantic data integration. By using LLMs as assistive generators of schema mappings rather than as black-box data transformation engines, the proposed approach decouples transformation logic from execution and enables scalable and transparent and semantically enhanced transformation pipelines. The grounding of this vision through ongoing work with the X3ML framework, demonstrates the practical feasibility of the proposed approach. Observations reveal strong indications that the time-consuming schema mapping process can be substantially accelerated. Overall, this vision opens opportunities for hybrid human-LLM workflows, in which language models support human experts, while established transformation frameworks ensure scalability in production settings. Looking to the future, several research directions stem from this work, such as methods to improve LLM-assisted mapping generation, the investigation of decomposing complex source data into smaller units that can be used for generating more fine-grained schema mappings, and methods for retrieving relevant existing mappings to support in-context generation.

¹https://github.com/ymark/x3ml_comparator

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] M. Fey, W. Hu, K. Huang, J. E. Lenssen, R. Ranjan, J. Robinson, R. Ying, J. You, J. Leskovec, Position: Relational deep learning-graph representation learning on relational databases, in: Forty-first International Conference on Machine Learning, 2024.
- [2] D. V. Assche, T. Delva, G. Haesendonck, P. Heyvaert, B. D. Meester, A. Dimou, Declarative rdf graph generation from heterogeneous (semi-) structured data: A systematic literature review, *Journal of Web Semantics* 75 (2023). doi:10.1016/j.websem.2022.100753.
- [3] Y. Marketakis, N. Minadakis, H. Kondylakis, K. Konsolaki, G. Samaritakis, M. Theodoridou, M. Doerr, X3ml mapping framework for information integration in cultural heritage and beyond, *International Journal on Digital Libraries* 18 (2017) 301–319. doi:10.1007/s00799-016-0179-1.
- [4] A. Dimou, M. V. Sande, P. Colpaert, R. Verborgh, E. Mannens, R. V. de Walle, Rml: A generic language for integrated rdf mappings of heterogeneous data, workshop on linked data on the web (ldow) 1184 (2014), 2014.
- [5] World Wide Web Consortium, R2rml: Rdb to rdf mapping language, 2012.
- [6] A. Nayak, H. P. Timmapathini, Llm2kb: Constructing knowledge bases using instruction tuned context aware large language models, international semantic web conference (iswc) workshop on knowledge base construction from pre-trained language models (2023), 2023.
- [7] S. Carta, A. Giuliani, L. Piano, A. S. Podda, L. Pompianu, S. G. Tiddia, Iterative zero-shot llm prompting for knowledge graph construction, 2023. doi:10.48550/arXiv.2307.01128. arXiv:2307.01128, arXiv preprint.
- [8] J. Frey, L. P. Meyer, N. Arndt, F. Brei, K. Bulert, Benchmarking the abilities of large language models for rdf knowledge graph creation and comprehension: How well do llms speak turtle?, 2023. doi:10.48550/arXiv.2309.17122. arXiv:2309.17122, arXiv preprint.
- [9] A. Sharma, X. Li, H. Guan, G. Sun, L. Zhang, L. Wang, K. Wu, et al., Automatic data transformation using large language model-an experimental study on building energy data, 2023 ieee international conference on big data (bigdata) (2023) 1824–1834, 2023. doi:10.1109/BigData59044.2023.10386931.
- [10] S. Ghazzai, D. Grigori, B. Benatallah, R. Rebai, Harnessing gpt for data transformation tasks, 2024 ieee international conference on web services (icws) (2024) 1329–1334, 2024. doi:10.1109/ICWS62655.2024.00160.
- [11] S. S. Norouzi, A. Barua, A. Christou, N. Gautam, A. Eells, P. Hitzler, C. Shimizu, Ontology population using llms, 2024. doi:10.48550/arXiv.2411.01612. arXiv:2411.01612, arXiv preprint.
- [12] T. Kenter, M. de Rijke, Short text similarity with word embeddings, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1411–1420. doi:10.1145/2806416.2806475.
- [13] X3ml Engine, 2025. URL: <https://github.com/isl/x3ml>.
- [14] A. Dimou, T. D. Nies, R. Verborgh, E. Mannens, P. Mechant, R. V. de Walle, Automated metadata generation for linked data generation and publishing workflows, in: Proceedings of the 9th Workshop on Linked Data on the Web, Montreal, Canada, 2016, 2016, pp. 1–10.
- [15] M. Doerr, The CIDOC CRM, an ontological approach to schema heterogeneity, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2005.
- [16] Y. Marketakis, M. Lintanff-Castel, Y. Tzitzikas, Using llms to automate schema mappings for rdf knowledge graphs construction, in: Poster in the 41st ACM/SIGAPP Symposium on Applied Computing, Thessaloniki, Greece, March 2026 (to appear), 2026.

```

<record>
  <id>pr-01</id>
  <name>Yannis</name>
</record>
<x3ml>
  <mappings><mapping>
    <domain>
      <source_node>//record</source_node>
      <target_node>
        <entity>
          <type>crm:E21_Person</type>
        </entity>
      </target_node>
    </domain>
    <link>
      <path>
        <source_relation>
          <relation>id</relation>
        </source_relation>
        <target_relation>
          <relationship>
            crm:P1_is_identified
          </relationship>
        </target_relation>
      </path>
      <range>
        <source_node>id</source_node>
        <target_node>
          <entity>
            <type>
              crm:E42_Identifier
            </type>
          </entity>
        </target_node>
      </range>
    </link>
  </mapping></mappings>
</x3ml>

```

Figure 2: Illustrative example of an XML source record, and the corresponding X3ML schema mapping.

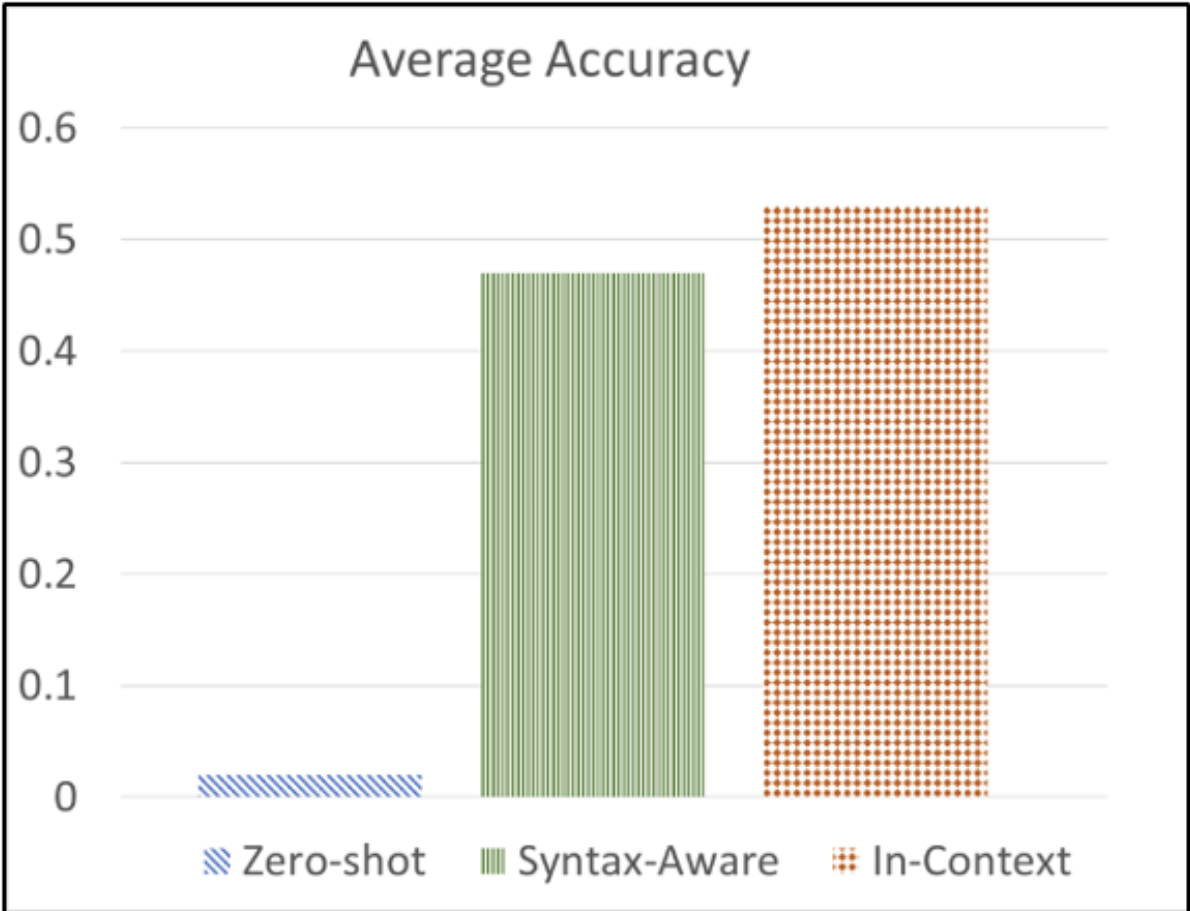


Figure 3: Aggregated accuracy of different prompting strategies.