Enhancing Ontology Engineering with LLMs: From Search to Active Learning Extensions

Ganna Kholmska
Jožef Stefan Institute
Ljubljana, Slovenia
anna.kholmska@gmail.com

Klemen Kenda
Jožef Stefan Institute
Ljubljana, Slovenia
klemen.kenda@ijs.si

Joze Rozanec
Jožef Stefan Institute
Ljubljana, Slovenia
joze.rozanec@ijs.si

Abstract

This paper explores the use of LLMs in ontology engineering within the HumAIne project, focusing on the discovery, analysis, and extension of ontologies in Data Mining, Machine Learning, and manufacturing. The methodology leverages fine-tuned prompts and combines LLMs with traditional tools like Protege for validation. A multi-LLM approach improved domain-specific concept coverage and reduced errors, though challenges remain in addressing deep domain-specific gaps and ensuring logical consistency.

Keywords

LLMs, Ontology Engineering, Active Learning, Data Mining, Machine Learning, Ontology Selection, Ontology Extension

1 Introduction

The HumAIne project, funded by the European Commission under the Horizon Europe program, aims to develop a platform integrating advanced AI paradigms such as Active Learning (AL), Neuro-Symbolic AI, Swarm Learning, and Explainable AI. This platform is designed to enhance human-AI collaboration in dynamic, unstructured environments, with applications spanning healthcare, manufacturing, finance, energy grids, and smart cities. Its primary goal is to support decision-making by combining human expertise with AI capabilities.

One of the project's key challenges is developing multiple ontologies that provide a structured framework for integrating domain-specific knowledge. This framework is essential for enhancing the clarity and reliability of AI-driven decisions, while ensuring adaptability across diverse applications. To construct these ontologies, we first explored publicly available ontologies relevant to the project's scope, then extended selected ones with concepts from HumAIne's AI paradigms, starting with Active Learning

However, manual ontology construction is a complex, resource-intensive process that requires expertise across multiple domains, collaboration among stakeholders. Ensuring modularity, reusability, and scalability adds to this complexity.

Information Society 2024, 7–11 October 2024, Ljubljana, Slovenia

https://doi.org/10.70314/is.2024.sikdd.28

Recent studies show that leveraging Large Language Models (LLMs) can streamline ontology construction by reducing manual effort and improving consistency and quality. For instance, [1] demonstrates semi-automatic knowledge graph construction using open-source LLMs, while [2] proposes methods for automatic concept hierarchy generation through LLM queries. Building on this research, this paper contributes a methodology that integrates LLMs with traditional tools like Protege to streamline the discovery, analysis, and extension of ontologies. By employing a multi-LLM approach, we address challenges in domain-specific concept identification and ensure more consistent, accurate results in ontology development for fields like Data Mining, Machine Learning, and manufacturing.

2 LLM-Assisted Search and Analysis of Domain Ontologies

Our experimentation with methodologies and tools for efficient web search and ontology analysis in Data Mining (DM), Machine Learning (ML), and manufacturing domains led to the development of the LLM-leveraging algorithm shown in Fig. 1. This algorithm uses carefully crafted prompts to guide LLMs in generating accurate, targeted queries. Before each step, the initial prompt is optimized through several iterations in a dialogue with the LLM to improve accuracy and relevance. Further details on the iterative query refinement process are provided in the Discussion section.

Step 1: Define the Search Objective. At this stage, LLMs like Bing Chat, Google's Bard, or ChatGPT with Web Browsing are employed to iteratively refine the search objectives initially formulated by the researcher, along with relevant keywords, phrases, and terms describing the ontologies or concepts of interest. For instance, our initial search objective for DM and ML ontologies was to "Find ontologies that offer up-to-date, detailed descriptions of the DM and ML domains, following best practices in ontology engineering." Keywords included "Active Learning" and "CRISP-DM standard."

Step 2: Formulate Search Queries Using LLMs. Based on the refined search objectives and keywords, and using a carefully crafted prompt, LLMs generate targeted search queries. These queries are fine-tuned through feedback or early search results to maximize relevance and accuracy. For example, for a DM ontology, the LLM generated queries such as "Data Mining ontology for semi-supervised machine learning," which were further refined before finalizing the query.

Step 3: Conduct Web Search. This step involves real-time browsing tools like Copilot in Microsoft Edge (GPT-4) and Perplexity AI to execute searches and identify relevant sources. Our study prioritized high-quality sources like ontology

repositories (e.g., BioPortal, OBO Foundry) and academic databases (Google Scholar, IEEE Xplore, ACM Digital Library).

It is important to acknowledge that LLM-driven web searches are generally confined to public repositories and a limited range of academic databases. As a result, proprietary or lesser-indexed ontologies may require manual exploration to ensure a more thorough search.

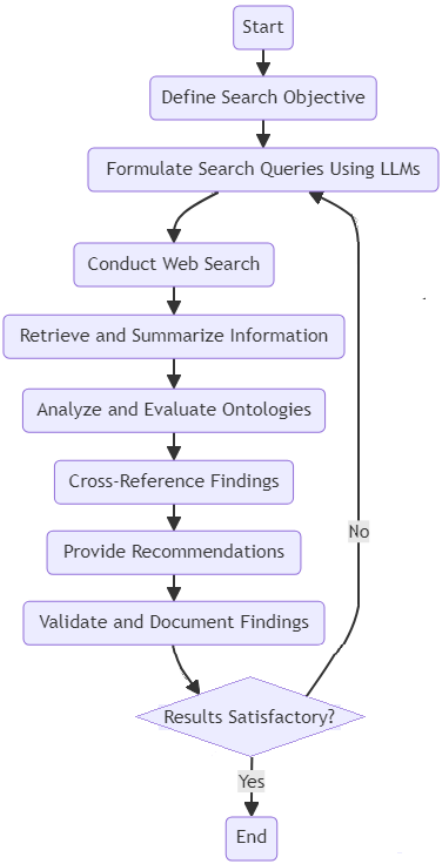

Figure 1: Key steps of LLM-leveraging algorithm

Step 4: Retrieve and Summarize Information. LLMs (Google Bard, Copilot (GPT-4), Perplexity AI) were employed to extract and summarize key information from ontology descriptions found in publications, technical papers, and repository documentation identified during the search. Using a specifically tuned prompt, LLMs extracted 11 characteristics for each of the 34 identified DM and ML ontologies. These characteristics included purpose, availability, ontology metrics, reused ontologies, software editors, representation language, and evaluation methodologies. This structured data, organized in table format, provided valuable insights into each ontology's scope, quality, and reusability. From these results, we selected 6 ontologies for further exploration, prioritizing comprehensive coverage of DM and ML concepts, adherence to ontology engineering best practices, and alignment with established standards in these domains.

Step 5: Analyze and evaluate ontologies. LLMs were further utilized to assess the relevance, content, and structure of the selected ontologies. In our study of DM and ML ontologies, LLMs such as GPT-4, which can process, explain, and generate OWL and RDF code, were used alongside ontology tools like Protege. This combination ensured that the ontologies addressed relevant concepts and aligned with frameworks like CRISP-DM. GPT-4 helped significantly in bridging the gap between textual descriptions and formal ontology representations.

Step 6: Cross-Reference and Compare Findings. LLMs with contextual understanding were employed to integrate and refine information from multiple sources. For this task, we used. Additionally, ChatGPT (GPT-4) categorized 65 manufacturing ontologies categorized 65 manufacturing ontologies, assessing them for relevance to process planning, standardization, industry adoption, interoperability, and support for advanced manufacturing concepts. Further exploration of the top 8 LLM-scored ontologies showed strong alignment with expert evaluations, but domain-specific tasks required carefully crafted prompts and human oversight for effectiveness.

Step 7: Provide Recommendations for Further Exploration. LLMs generated recommendations for the most suitable ontologies or areas for additional research based on the previous step's results. This includes identifying underexplored concepts and areas needing further investigation.

Step 8: Validate and Document Findings. The findings were manually validated for accuracy and relevance, then systematically documented. ChatGPT (GPT-4) was used to summarize and structure the documentation.

Step 9: Iterate and Refine Search (if needed). When results were too broad or irrelevant (e.g., Active Learning misinterpreted as an educational method), we refined the search prompt by adding more context.

By using this LLM-based algorithm, we conducted comprehensive web searches and extracted relevant information to identify the most suitable ontologies for the HumAIne project. In the DM and ML domains, we selected the OntoDM suite (OntoDM-Core, Onto-KDD, and OntoDT). For the manufacturing domain, we identified the Industrial Ontologies Foundry Core (IOF Core) as the best fit.

3 LLM-Assisted Ontology Extension with Active Learning Concepts

Integrating Active Learning (AL) into an ontology requires extending it with new classes, properties, and relationships representing key AL concepts. While traditional methods of building and extending ontologies are well-documented, we leveraged GPT-4 for this task using iteratively refined prompts (see Discussion section). This section outlines how LLMs, particularly GPT-4, were used to extend the IOF Core ontology with AL concepts.

Step 1: Define the Problem and Objectives. Through iteratively refined prompts, LLMs formulated clear objectives, specifying the domain (e.g., manufacturing) and key concepts (e.g., Active Learning). These outputs were used to guide further steps, with LLMs leveraging contextual understanding, knowledge synthesis, and language generation to suggest relevant AL applications such as adaptive scheduling. Queries like "How can Active Learning improve adaptive scheduling in manufacturing?" generated valuable insights into potential use cases. where AL would be most beneficial.

Step 2: Analyze the Ontology to be Extended. By combining Protege's visualization and navigation tools with GPT-4's ability

to process textual and machine-readable data (e.g., OWL/RDF), we thoroughly examined the IOF Core ontology structure and identified areas for introducing AL concepts. For example, GPT-4 helped uncover key classes like “Process,” “Resource,” and “PerformanceMetric” within IOF Core, highlighting relevant properties for AL integration. Queries such as "What aspects of IOF Core can benefit from AL integration?" and "What key concepts are missing from the IOF Core ontology for integrating Active Learning in manufacturing?" guided us in identifying areas for improvement, including handling uncertainty and adjusting dynamic processes.

Step 3: Identify Active Learning Concepts. The main tasks of this step and the role of LLMs in supporting each task are summarized in the Table 1:

Table 1: LLMs applications for Identifying AL Concepts

Task	LLM Application	Example Output
1. Identify fundamental AL concepts	Use LLMs to generate a list of core AL strategies and techniques	Concepts like“Uncertainty sampling,” “Query-by-committee”
2. Extract domain-specific AL concepts	Query LLMs about AL in specific industrial contexts	Concept like "Query Efficiency" in decision-making for manufacturing
3. Mine AL concepts from literature	Process academic papers, reports to extract relevant AL terms	Concepts like “Stream-base selective sampling” from papers on AL in manufacturing
4. Assign properties to new classes	Generate properties for AL ontology classes	QueryStrategy class properties: “hasuncertaintySampling” “queryByCommittee”
5. Refine and validate terminology	Ensure definitions, resolve overlaps	Refined and validated terms based on domain-specific standards

By prompting, LLMs generated nearly 200 fundamental AL concepts, structuring them into a hierarchy by leveraging their vast training data. Additionally, LLMs helped generate definitions, assisting in verifying and refining concepts. However, after a point, LLMs began repeating concepts or producing less relevant terms. LLMs were also effective in generating domain-specific concepts through targeted queries. For instance, querying AL in manufacturing led to concepts like "uncertainty management" and "query efficiency." More specialized concepts required extraction from academic papers, which were cross-referenced with existing standards in DM, ML, and manufacturing (e.g., CRISP-DM, IEEE 7000 Series, ISA-95, ISO 15531). Ontology learning tools like Text2Onto and OntoLearn were combined with LLMs for cross-verification.

*Step 4: Develop Ontology Extensions.*LLMs helped create AL-related classes, properties, and relationships based on the identified concepts, using OWL-compliant syntax (see Fig. 2). By combining GPT-4’s knowledge synthesis with Protege’s structural reasoning and consistency checking, we improved the efficiency and accuracy of reviewing, debugging, and validating OWL code.

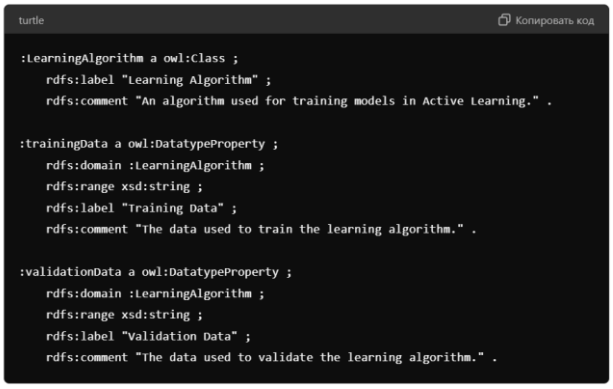

Figure 2: Screenshot of LLM-generated code defining the “LearningAlgorithm” class with properties “trainingData” and “validationData”

Step 5: Ensure Semantic Consistency. LLMs, such as GPT-4, assisted in ensuring semantic consistency by reviewing new and existing ontology elements and suggesting how new concepts could align with the existing framework. For example, an LLM suggested how an AL “QueryStrategy” class fits within the IOF Core ontology.

Example Prompt: " Review the new QueryStrategy class and suggest how it can align with the existing classes in IOF Core."

LLM Output: The QueryStrategy class aligns with decision-making aspects of the Process concept. Strategies such as “UncertaintySampling,” “QueryByCommittee,” “ExpectedModelChange,” and “ExpectedErrorReduction” can be viewed as specialized decision-making processes within the broader process framework of IOF Core.

However, LLMs cannot guarantee logical consistency and face limitations in handling complex relationships, making it necessary to use ontology reasoners, such as Protege or HermiT, to perform consistency checks.

Step 6: Map to Existing Ontologies. LLMs, such as GPT-4, assist in generating initial mapping suggestions by analyzing similarities in definitions, relationships, and properties between new and existing concepts. This involves creating explicit relationships like “owl:sameAs,” “owl:equivalentClass”, and “owl:equivalentProperty”.

Example LLM Output:

:FeedbackMechanism a owl:Class ;
owl:equivalentClass :ControlSystem ;
rdfs:label "Feedback Mechanism" ;
rdfs:comment "Mechanisms that provide feedback in Active Learning to control systems."

While LLMs are effective in identifying high-level similarities, they may face challenges with complex or domain-specific relationships, requiring further refinement. Although we didn’t encounter these issues during our initial work extending IOF Core with AL concepts, we used Protege’s alignment plug-ins to refine LLM-generated mappings. For more complex mappings, tools like AgreementMaker or COMA can further refine the suggestions.

Step 7: Prototype and Test. LLMs, such as GPT-4, were prompted to generate validation scenarios, competency questions, and SPARQL queries based on the integrated AL concepts. For instance, a prompt like "Suggest validation scenarios for adaptive scheduling with Active Learning" helped us produce realistic test cases, including prototype code, descriptions of initial setup, process flows, validation steps, and queries based on newly integrated concepts.

SPARQL queries generated by LLMs were executed using Protege with SPARQL plugins to assess the ontology's ability to retrieve relevant information and answer competency questions.

However, some LLM-generated scenarios revealed limitations in domain-specific knowledge, resulting in generic outputs that required refinement. Additionally, LLMs struggled with modeling intricate relationships or complex data retrieval conditions, making human oversight essential for ensuring accuracy and thorough testing.

Step 8: Iterative Refinement. Following initial prototyping and testing, we gathered feedback from domain experts and users to further refine the ontology. Validation reports were uploaded to AskPDF Research Assistant (GPT-4), where LLMs reviewed the reports, extracted key improvement suggestions, and refined task lists. The LLM provided insights into areas where ontology relationships or properties required adjustments and identified additional concepts that might have been overlooked.

Step 9: Document and Disseminate. LLMs like ChatGPT or Bard were instrumental in generating comprehensive documentation, including details on the ontology extensions. Additionally, LLMs contributed to drafting technical reports and research papers.

Using this methodology, we successfully extended the IOF Core ontology with Active Learning (AL) concepts. Future stages of the HumAIne project will focus on further validation and refinement, particularly during pilot case implementations.

4 Discussion

This study highlights LLMs' potential in ontology engineering by reducing manual effort and increasing efficiency. LLMs rapidly identified key ontologies like OntoDM and IOF Core and generated structured classes, properties, and relationships, reducing the need for manual OWL/RDF code generation and concept mapping. However, LLMs face challenges in domain-specific precision, requiring human oversight to refine outputs and address nuances in specialized fields. While tools like Protege excel at ensuring logical consistency, LLMs offer dynamic capabilities for generating new concepts and relationships. Despite these advantages, traditional tools like AgreementMaker and COMA are still necessary to refine and validate LLM-generated mappings.

One strategy to mitigate LLM limitations was iterative prompt engineering. We refined prompts for ontology search and extension tasks through multiple cycles of improvement. These cycles, with LLMs like GPT-4, involved clarifying questions, refining queries, and generating more focused outputs. Initial prompt for starting the cycle can be the following:

"Your role is my Prompt Creator. Your goal is to craft the best possible prompt for my needs. The prompt will be used by you, [LLM's name]. I want to write about: [keyword/topic]. Based on my input, you will now generate 3 sections. a) Revised prompt (clear, concise, and easily understood by you), b) Suggestions (on what details to include in the prompt to improve it), and c) Questions (ask any relevant questions to improve the prompt). We will continue this iterative process with me providing additional information to you and you updating the prompt until it's complete."

After 4-5 cycles, the prompts were highly optimized, ensuring relevant outputs. This refinement process reduced inconsistencies and improved LLM-generated content across both search and extension phases.

We integrated multiple LLMs, including Bing Chat (GPT-4), Google's Bard, and Perplexity AI, to cross-validate outputs, reducing errors and refining results. This ensured consistency in LLM-generated ontologies and mappings.

To evaluate this multi-LLM approach, we propose the following metrics: Inter-Model Consistency (measures alignment between LLM outputs). Error Rate Reduction (Tracks how often one LLM corrects another's errors),.Coverage of Relevant Concepts (assesses LLMs' ability to capture domain-specific concepts). Although these metrics provide a framework, formal measurements are yet to be implemented.

Future stages will involve applying these metrics to validate outputs and testing extended ontologies in real-world applications. This hybrid method combines LLMs and traditional tools, ensuring both efficiency and accuracy in scalable ontology development.

5 Conclusions

This study demonstrates how LLMs can streamline ontology engineering by automating the search, analysis, and extension of domain-specific ontologies. Leveraging multiple LLMs, we successfully identified and extended key ontologies, including OntoDM and IOF Core, for the HumAIne project, improving efficiency in generating classes, properties, and relationships.

While LLMs significantly enhance the process, they face challenges in domain-specific precision and require human oversight, particularly for complex relationships. Traditional tools like Protege and ontology reasoners remain critical for ensuring logical consistency and validation.

Future work will focus on refining these extended ontologies through real-world pilot tests and applying evaluation metrics to LLM-generated outputs. This hybrid approach, combining LLM automation with traditional validation tools, offers a scalable solution that balances efficiency with the need for human expertise.

Acknowledgments

This work was supported by the European Commission under the Horizon Europe project HumAIne, Grant Agreement No. 101120218.

References

[1] Kommineni, Vamsi Krishna, Birgitta König-Ries and Sheeba Samuel. "From human experts to machines: An LLM supported approach to ontology and knowledge graph construction." *ArXiv* abs/2403.08345 (2024): n. pag. DOI: https://doi.org/10.48550/arXiv.2403.08345

[2] Funk, Maurice, Simon Hosemann, Jean Christoph Jung and Carsten Lutz. "Towards Ontology Construction with Language Models." ArXiv abs/2309.09898 (2023): DOI: https://doi.org/10.48550/arXiv.2309.09898