Large Language Models for Ontology Engineering: A Systematic Literature Review

Journal Title
XX(X):1–45

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Jiayi Li[1], Daniel Garijo[1], María Poveda-Villalón[1]

Abstract

Ontology engineering (OE) is a complex task in knowledge representation that relies heavily on domain experts to accurately define concepts and precise relationships in a domain of interest, as well as to maintain logical consistency throughout the resultant ontology. Recent advancements in Large Language Models (LLMs) have created new opportunities to automate and enhance various stages of ontology development. This paper presents a systematic literature review on the use of LLMs in OE, focusing on their roles in core development activities, input-output characteristics, evaluation methods, and application domains. We analyze 30 different papers to identify common tasks where LLMs have been applied, such as ontology requirements specification, implementation, publication, and maintenance. Our findings indicate that LLMs serve primarily as ontology engineers, domain experts, and evaluators, using models such as GPT, LLaMA, and T5 to process heterogeneous inputs (such as OWL ontologies, text, competency questions, etc.) to generate task-specific outputs (such as examples, axioms, documentation, etc.). Our review also observed a lack of homogenization in task definitions, dataset selection, evaluation metrics, and experimental workflows. At the same time, some papers do not release complete evaluation protocols or code, making their results hard to reproduce and their methods insufficiently transparent. Therefore, the development of standardized benchmarks and hybrid workflows that integrate LLM automation with human expertise will become an important challenge for future research.

Keywords

Large Language Models,Ontology Engineering, Ontology Development,Systematic Review Survey

[0] [1]Ontology Engineering Group, Universidad Politécnica de Madrid, Boadilla del Monte, 28660, Spain
[0]
Corresponding author:
Jiayi Li, Ontology Engineering Group, Universidad Politécnica de Madrid, Boadilla del Monte, 28660, Spain
[0]Email: li.jiayi@upm.es

1 Introduction

For more than a decade, Knowledge Graphs have become a key technology for representing, using and sharing open knowledge in a wide range of domains and applications Hogan et al. (2021). To give these rich datasets formal, machine-readable structure and semantics Patel and Debnath (2024); Glauer et al. (2024), ontologies are employed to define domain-specific concepts, relationships, constraints, and logical rules De Vergara et al. (2004); Patel and Debnath (2024); Glauer et al. (2024). Ontologies are typically encoded in the W3C Web Ontology Language (OWL) Staab et al. (2004), and have been shown to work effectively to integrate, validate, and reason with data in KGs Krötzsch and Thost (2016).

Ontology engineering (OE) is the process of developing formal knowledge representations (i.e., ontologies) to describe aspects of reality for specific purposes Salamon and Barcellos (2022). Despite the availability of structured methodologies such as Linked Open Terms (LOT) Poveda-Villalón et al. (2022), NeOn Suárez-Figueroa et al. (2012), the "Ontology Development 101" guide Noy and McGuinness (2001), etc., ontology development remains a complex, time-consuming, and error-prone activity Gangemi and Presutti (2009); Saeedizade and Blomqvist (2024). It demands deep domain expertise, careful conceptual modeling, extensive collaboration among stakeholders, and precise alignment with intended use cases Poveda-Villalón et al. (2022).

With the development of Artificial Intelligence (AI), significant advancements have been made in Large Language Models (LLMs) to show remarkable advances in capturing complex language patterns in different knowledge domains Doumanas et al. (2024). In recent years, LLMs have emerged as an innovative technology for OE. Research efforts have explored their potential to assist developers in various tasks, including generating and refining ontologies from text, aligning concepts with existing taxonomies, and automatically detecting syntax errors in ontologies, among others Garijo et al. (2024).

Despite the promise of LLMs for OE, several key research gaps remain. Many studies have claimed that LLMs are useful for ontology development tasks Lo et al. (2024); Joachimiak et al. (2024); Lippolis et al. (2025, 2024); Ciatto et al. (2025), but do not clearly distinguish the specific development phases where LLMs provide the most value. In addition, little is known about the specific roles LLMs can assume, the types of inputs and outputs required by them, the necessity and extent of human involvement, and the experimental setups, including datasets used, evaluation metrics, and reproducibility considerations used to validate their effectiveness. Furthermore, while LLMs are increasingly applied in various domains, few studies systematically address domain-specific challenges or necessary model adaptations.

Although recent surveys have offered valuable overviews of LLMs in OE Perera and Liu (2024); Garijo et al. (2024), a detailed analysis focusing specifically on ontology development activities remains limited. A systematic understanding of how LLMs contribute to different phases of ontology development, along with a critical assessment of their capabilities and limitations, is essential for guiding future research and fostering their successful integration into OE workflows.

To address these gaps, this study conducts a comprehensive and systematic review of how LLMs are employed in ontology engineering. Our work extends our initial overview Garijo et al. (2024) with the following objectives:

1. Identify the ontology development tasks where LLMs have been applied.

2. Analyze how LLM-based approaches contribute to ontology development, focusing on their roles, model types, inputs, outputs, and the role of human participants in interactive workflows.

3. Examine how LLM performance is assessed in ontology development by identifying experimental datasets, evaluation methods, and reported performance results.

4. Explore the application domains where LLMs have been effectively utilized for ontology development.

We conduct our review following the systematic methodology proposed by Kitchenham et al. Kitchenham et al. (2009), ensuring a rigorous and reproducible analysis. We also make publicly available the complete corpus of resources used to generate or evaluate different OE tasks at our GitHub repository[1], along with a preserved version archived on Zenodo (DOI: 10.5281/zenodo.15313672).

The remainder of this article is organized as follows. Section 2 presents background information on OE and LLM technologies. Section 3.1 outlines our research objectives and key questions. Section 3 describes our data collection and analysis methods. Section 4 presents the results and discusses key insights. Sections 5 and 6 conclude the study and suggest directions for future research. Also, additional Section 7 supporting materials are provided in the appendix.

2 Background

In this section, we briefly introduce the main ontology development tasks identified in the literature and provide an overview of the recent evolution of LLMs.

2.1 Ontology Development Tasks

Ontologies are formal and explicit specifications of shared conceptualizations Studer et al. (1998), enabling structured knowledge representation Dimitropoulos and Hatzilygeroudis (2024) and facilitating semantic interoperability across systems and applications Bittner et al. (2005); Tan et al. (2024).

Ontology Engineering (OE) Gómez-Pérez (1999) provides the methodologies and tools necessary for constructing domain-specific and application-specific ontological models. An Ontology Engineering Methodology (OEM) outlines a structured set of phases, processes, and tasks to systematically guide the development process Kotis et al. (2020).

Traditional methodologies, such as METHONTOLOGY Fernández-López et al. (1997), On-To-Knowledge Staab et al. (2001), DILIGENT Pinto et al. (2004), and the "Ontology Development 101" guide Noy and McGuinness (2001), have significantly contributed to the formalization of OE practices. However, they typically follow step-by-step workflows that may not fully address modern requirements such as reuse, collaboration, and interoperability. The NeOn methodology Suárez-Figueroa et al. (2012) introduced a more dynamic and flexible approach, emphasizing the creation of interconnected ontology networks through mechanisms like import, versioning, mapping, and modularization.

To consider a basic group of activities usually carried out during ontology development we follow the Linked Open Terms (LOT) methodology Poveda-Villalón et al. (2022) general workflow as it includes ontology publication and maintenance phases. However, other activities not defined in detail in LOT may appear in the reviewed works. In order to address these cases we also consider the NeOn glossary of activities Suárez-Figueroa and Gómez-Pérez (2008). It should be noted that both LOT and NeOn define

[1] https://github.com/oeg-upm/llm4oe-slr

more activities than the ones listed below, however, we include in this section only those activities found in the reviewed papers.

1. **Ontology requirements specification phase:** Gathering requirements is related to the specific ontology goals, domain, and technical constraints Suárez-Figueroa et al. (2009). From the activites defined for this phase, in the analyzed papers the following activities are addressed:

    - *Functional requirements writing:* Specifies the functionalities the ontology must support. It should be noted that this activity refers to writing the functional requirements in natural language text. This may occur in the form of competency questions or affirmative sentences in natural language.
    - *Competency Question reverse engineering:* Involves generating CQs that an ontology must answer, using the ontology itself as input. Although not explicitly covered in the LOT framework, this activity appears in several studies (Alharbi et al. (2024a); Keet et al. (2019)) and aligns with NeOn Ontological Resource Reverse Reengineering Suárez-Figueroa et al. (2012).
    - *Requirement Formalization:* This activity consist in translating functional requirements into formal, machine-readable specifications.

2. **Ontology implementation phase:** Building the ontology using formal languages (e.g., OWL, RDF) based on collected requirements. Key sub-activities include:

    - *Conceptualization:* Structuring domain knowledge into concepts and relationships.
    - *Encoding:* Formalizing conceptual models into machine-readable formats (e.g., Turtle, RDF/XML, etc.).
    - *Evaluation:* Validating the ontology against competency questions and domain needs.
    - *Matching:* This activity's definition is taken from NeOn which literally reads "the activity of finding or discovering relationships or correspondences between entities of different ontologies or ontology modules" Suárez-Figueroa and Gómez-Pérez (2008).

3. **Ontology publication phase:** Making the ontology accessible both as human-readable documentation and machine-readable files. This phase includes, among others not found in the reviewed papers as the actual online publication, the following activity:

    - *Documentation:* Generating human-oriented documentation usually consisting, but not limited to HTML web pages, diagrams, examples of use, etc.

4. **Ontology maintenance phase:** Updating the ontology based on bug reports, improvements, and new requirements throughout its lifecycle. This includes:

    - *Bug Detection:* Identifying and reporting errors or inconsistencies.

2.2 A Brief History of Large Language Models

LLMs are AI systems able to generate coherent and contextually relevant language outputs Goyal et al. (2022) that have demonstrated remarkable performance across tasks like text generation Goyal et al. (2022), question answering Nakano et al. (2021), translation Brown et al. (2020b), summarization Xie et al. (2023), and sentiment analysis Kheiri and Karimi (2023). LLms are trained on large amounts of textual data, and are built predominantly on deep learning architectures such as transformers Vaswani et al. (2017).

The evolution of LLMs began with foundational advancements in sequential data processing. Rumelhart et al. Rumelhart et al. (1986) introduced recurrent neural networks (RNNs), which were later enhanced by the Long Short-Term Memory (LSTM) model developed by Hochreiter and Schmidhuber Hochreiter and Schmidhuber (1997), significantly improving long-range dependency modeling Mienye et al. (2024). The release of the Generative Pre-trained Transformer (GPT) by OpenAI in 2018 marked a pivotal moment. Subsequent iterations (GPT-2, GPT-3, GPT-3.5) Brown et al. (2020a); Radford et al. (2019) demonstrated increasingly sophisticated generative capabilities Touvron et al. (2023b); DeepMind (2023). GPT-3, for instance, was trained on 45TB of data and contained 175 billion parameters. In 2023, Meta introduced LLaMA Touvron et al. (2023a), an open-source LLM trained on 1.4 trillion tokens across multiple model sizes. Since then, models such as Google Gemini Team et al. (2024), OpenAI's GPT-4 OpenAI et al. (2024), Meta's LLaMA2 Touvron et al. (2023b), and LLaMA3 Grattafiori et al. (2024) have further advanced the field. These models exhibit state-of-the-art performance in reasoning Huang and Chang (2022); Wei et al. (2022), code generation Vaithilingam et al. (2022); Singh et al. (2023); Jiang et al. (2024), and multimodal tasks Zhang et al. (2023); Wu et al. (2023); Zhang et al. (2024a), driven by larger datasets and increasingly sophisticated architectures. Their ongoing evolution continues to expand the application landscape for AI-driven systems across diverse domains Johnsen (2025).

3 Research Methodology

To achieve our research objectives, we conducted a systematic literature review following Kitchenham and Charters methodology Kitchenham et al. (2009): Section 3.1 defines the research questions (RQs) of our study, Section 3.2 describes the selection of data sources, Section 3.3 presents the search strategy, Section 3.4 explains the filtering criteria, and Section 3.5 details data extraction and synthesis. The following subsections describe each step.

3.1 Research Questions

Our study investigates how LLMs have been adapted for ontology development by systematically reviewing existing approaches to understand their capabilities and limitations. We formulate the following RQs to guide our review:

RQ1 What are the key activities in ontology development where LLMs have been applied?

RQ2 How do LLM-based approaches support different ontology development activities?

- **RQ2.a** What roles do LLMs play in these activities?
- **RQ2.b** What types of LLMs are used?

RQ2.c What are the typical inputs to the LLMs?

RQ2.d What outputs are generated by the LLMs?

RQ2.e What are the roles of humans involved in these activities (e.g., domain experts, ontology engineers)?

RQ3 How is the performance of LLMs in ontology development evaluated?

RQ3.a Are there evaluation experiments reported?

RQ3.b What datasets are used in the evaluations?

RQ3.c What evaluation methods are adopted (e.g., qualitative, quantitative, or hybrid)?

RQ3.d What metrics (e.g., F1 score, recall) are used, and what are the reported performance results?

RQ4 What are the main application domains where LLMs have been applied in ontology development?

3.2 Source Libraries

During this phase, we conduct a systematic search across open-access digital libraries to ensure comprehensive coverage of the area under investigation Vieira and Gomes (2009). We selected Google Scholar Boell and Cecez-Kecmanovic (2014), Web of Science, and Scopus Carrera-Rivera et al. (2022) for their broad multidisciplinary reach, along with the ACM Digital Library and IEEE Xplore to specifically cover the computer science domain Hull et al. (2008). The selected sources and their corresponding access points are: Google Scholar[2], Web of Science[3], Scopus[4], ACM Digital Library[5], and IEEE Xplore[6].

3.3 Search Strategy

The selection of primary studies depend on the following inclusion and exclusion criteria:

1. **Publication Time Frame:** We focus on papers from 2018 to 2024 to capture the latest advances in ontology development using LLMs. The year 2018 marks a key milestone with the introduction of the LLM keyword in some papers Radford et al. (2018); Brown et al. (2020b), which paved the way for rapid progress in LLMs.

2. **Peer-Review Status:** Selecting peer-reviewed papers ensures rigorous expert evaluation, enhancing the high quality, credibility, and reliability of our findings Kelly et al. (2014).

3. **Language:** We focus on papers, books, and book chapters published in English for accessibility and consistency.

4. **Search Keywords:** Our search focus on two categories of terms:

[2] https://scholar.google.com
[3] https://www.webofscience.com
[4] https://www.scopus.com
[5] https://dl.acm.org
[6] https://ieeexplore.ieee.org

(a) Semantic-Related Terms (SR): Keywords related to semantic technologies, such as ontolog*, ontology development and vocabulary.

(b) Model-Related Terms (MR): Keywords associated with large language models, including Language Model, LM, and LLM*

The particularities of each source were considered during the review. Logical operators (OR, AND) combined terms into search strings, such as ('ontolog*' OR 'ontology development') AND ('LM' OR 'LLM*'), applied to meta-fields searched from Section 3.2. Depending on each source, the search strings were tailored to content, title, abstract, and keywords.

3.4 Filtering Process

In this step, we apply our search criteria to the selected library sources through a two-stage filtering process.

1. **Automated Filtering:** We first applied automated filters based on the predefined search standards and removed duplicate papers by matching their titles.
2. **Manual Filtering:** To further ensure relevance, we conducted a multi-stage manual review, comprising the following steps:
   (a) **Title Screening:** We initially reviewed the titles of the retrieved papers to eliminate papers that were clearly unrelated to our research topic.
   (b) **Abstract Screening:** For the remaining papers, we examined the abstracts to assess their alignment with our research objectives. Only peer-reviewed papers that explicitly addressed the role of LLMs in ontology development were retained.

3.5 Data Extraction

To extract relevant information, we aligned the data extraction process with the RQs defined in Section 3.1. Since a single article may involve multiple ontology development activity experiments, each activity was recorded as a separate row in the dataset. The complete dataset is publicly available in our open repository at https://github.com/oeg-upm/llm4oe-slr.

Specifically, we extracted the following information from each entry:

- **Article metadata:** Publication title, authors, publication year, peer-reviewed status, and language.
- **Ontology Activity (RQ1):** The ontology development activity supported by LLMs and its definition (if provided).
- **LLM Technology (RQ2):** Role of the LLM in the activity, type of LLM used, inputs provided to the LLM, outputs generated, whether human-in-the-loop involvement was present (Yes/No), role of the human (e.g., ontology engineers and others), and tasks performed by human participants.
- **Performance Evaluation (RQ3):** Existence of evaluation experiments, links to experiments (if available), datasets used, dataset types, baselines compared, evaluation methods (quantitative, qualitative, or hybrid), metrics applied (e.g., F1 score, recall), and performance results, including whether humans participated in the evaluation.

- **Application Domains (RQ4):** Domains where LLMs were applied, such as healthcare, education, and finance.

4 Search Results

Our initial search yielded 11,985 results, reduced to 5,275 unique papers after duplicate removal. Title screening narrowed this to 204 papers, and abstract screening further shortlisted 38 peer-reviewed papers related to LLM use in ontology development. After excluding two review papers Garijo et al. (2024); Perera and Liu (2024) and six studies that primarily used LLMs for knowledge extraction or ontology population rather than for ontology design or structural development Usmanova and Usbeck (2024); Mukanova et al. (2024); Sahbi et al. (2024); Tian et al. (2023); Funk et al. (2023), 30 papers were ultimately retained for analysis as showin in Figure 1. In the following subsections, we present and discuss the findings for each of our RQs.

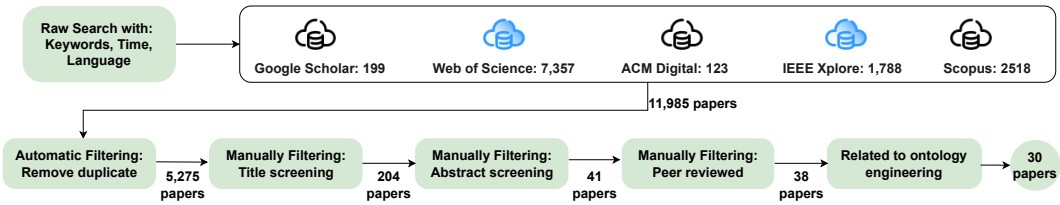

Figure 1. Paper selection process based on our methodology. From 11,985 papers retrieved across five libraries, 30 papers related to our LLM-based OE tasks were selected after duplicate removal and manual filtering.

4.1 RQ1: What are the key activities in ontology development where LLMs have been applied?

The first step in our study is to analyze in which ontology engineering activities are LLMs applied. Table 1 compiles the activities addressed in each of the analyzed approaches including the input and outputs provided to the LLM for each activity. A paper may address more than one ontology development activity, and therefore the same paper may lead to multiple rows in the table. As shown in Figure 2, most of the attention is focused on activities related to ontology implementation tasks (encoding, conceptualization, matching or evaluation) as well as the generation of requirements. Each approach is summarized in the following section grouping them by OE activity addressed.

4.1.1 Ontology requirements specification

In the task of **functional requirements specification**, Fathallah et al. (2024a) proposed a method leveraging LLMs such as GPT-3.5, LLaMA, and PaLM via zero-shot prompting to generate ontology requirements from natural language texts and CQs, within the framework of the NeOn-GPT methodology, using a wine ontology as a case study.

CQ reverse engineering has received growing attention by creating CQs directly from ontologies. Alharbi et al. (2024a) developed a pipeline that parses existing ontologies to extract relevant information, which is then used to instantiate prompts for the automated generation of candidate CQs. Rebboud et al.

Table 1. Summary of ontology development phases, tasks, resources, inputs, and outputs supported by LLMs. For studies applying LLMs at multiple workflow stages (e.g., Doumanas et al. (2024), Kholmska et al. (2024)), we list each task to separately to capture their distinct contributions.

Phase	Task	Resource	Inputs	Outputs
Requirements specification	Functional requirements writing	Fathallah et al. (2024a)	Natural language text	Natural language text
		Antia and Keet (2023)	Natural language text	CQs
	CQ reverse engineering	Rebboud et al. (2024a)	Ontologies	CQs
		Alharbi et al. (2024a)	Triples	CQs
		Ciroku et al. (2024a)	KGs	CQs
		Rebboud et al. (2024b)	Ontologies	CQs
		Alharbi et al. (2024b)	Triples	CQs
	Requirement formalization	Rebboud et al. (2024a)	Ontologies and CQs	Queries
		Tufek et al. (2024)	Natural language text or CQs	SPARQL Queries
		Kholmska et al. (2024)	Concepts	SPARQL Queries
Ontology implementation	Conceptualization	Rebboud et al. (2024a)	CQs	Ontologies
		Bischof et al. (2024)	Natural language text	Terms
		Goyal et al. (2024)	Natural language text	Binary decision
		Coutinho (2024)	Natural language text	Summarization
		Kholmska et al. (2024)	Step 2:Natural language text Step 3:Natural language text	Step 2: Classes Step 3: Concepts
		Dong et al. (2024)	Natural language text, Ontologies	Natural language text
		Babaei Giglou et al. (2023)	Task A: Natural language text, lexical term Task B: Natural language text Task C: Natural language text	Task A: Term type Task B: Binary decision Task C: Binary decision
		Toro et al. (2024)	Term	JSON or YAML
		Pisu et al. (2024)	Nature language text	Relationships
	Encoding	Doumanas et al. (2024)	Phase 1: Natural language text Phase 2: Domain documents Phase 3: Natural language text and CQs	Phase 1: Ontologies Phase 2: Ontologies Phase 3: Ontologies
		Fathallah et al. (2024a)	Natural language text	CQs, Triples and Ontologies
		Caufield et al. (2024)	Natural language text	Ontologies
		Eells et al. (2024)	Natural language text	Natural language text and RDF
		Saeedizade and Blomqvist (2024)	CQs	Ontologies
		Mateiu and Groza (2023)	Natural language text	Axioms
		Tang et al. (2023)	Natural language text	Ontologies, JSON and Triples
		da Silva et al. (2024)	Natural language text, Ontologies	Ontologies
	Ontology matching	Zamazal (2024)	Natural language text and verbalized candidates	Binary decision
		Kholmska et al. (2024)	Step 4: Natural language text Step 6: Concepts, Ontologies, Natural language text	Step 4: Documentation Step 6: Mapping
		Hertling and Paulheim (2023)	Ontologies and Natural language text	Mapping
		He et al. (2023)	Natural language text	Binary decision
		Norouzi et al. (2023)	Natural language text	Mapping
	Ontology evaluation	Tsaneva et al. (2024)	Natural language text	Axioms
		Kholmska et al. (2024)	Step 5: Ontologies	Step 5: Natural language text
		Fathallah et al. (2024a)	Natural language text	Ontologies and Axioms
		Zhang et al. (2025)	Ontologies and CQs	Binary decision
Ontology publication	Ontology documentation	Rebboud et al. (2024a)	Ontologies	Documentation
		Kholmska et al. (2024)	Step 9: Ontology Extensions, Natural language text	Step 9: Documentation
		Fathallah et al. (2024a)	Natural language text, Ontologies	Documentation
		Giri et al. (2024)	Terms	Documentation
Maintenance	Bug issue	Kholmska et al. (2024)	Step 8: Natural language text	Step 8: Natural language text

(2024a) introduced a benchmarking strategy that include generating CQs from ontologies, using tools such as LangChain and Ollama.

Several additional contributions enrich this area. For example, Ciroku et al. (2024a) introduced RevOnt, a system for extracting CQs from knowledge graphs. Rebboud et al. (2024b) conducted a feasibility study comparing LLM-generated CQs with ground-truth examples. Antia and Keet (2023) presented AgOCQs, a pipeline that combines a text corpus with CQ templates with NLP techniques to generate CQs.

Once requirements and CQs are established, **requirement formalization** can automate transfer CQs into executable queries, a crucial step in ontology development. Kholmska et al. (2024) investigate the

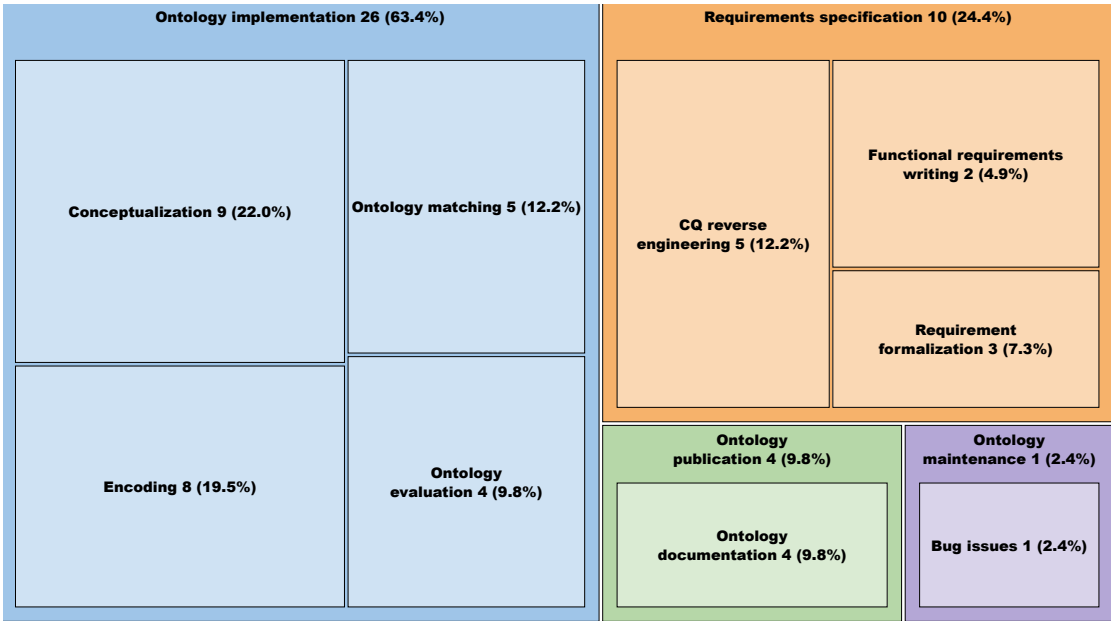

Figure 2. Distribution of LLM-supported tasks across ontology development phases based on 41 experiments from 30 papers. Numbers represent the total tasks identified for each phase, and percentages indicate their proportion relative to all tasks. Most tasks focus on ontology implementation (26 studies, 63.4%), followed by requirements specification (10 studies, 24.4%), publication (4 studies, 9.8%), and maintenance (1 study, 2.4%).

role of LLMs (e.g., ChatGPT, Bard, Perplexity AI) in OE with active learning, demonstrating their ability to generate SPARQL queries from CQs. Similarly, Rebboud et al. (2024a) benchmarked LLM-generated ontology-aligned queries, evaluating them using structural metrics like Tree Edit Distance. Their results show that LLMs are able to capture ontology structure and user intent. Further supporting this, Tufek et al. (2024) successfully automated SPARQL query generation from natural language requirements, showing that LLMs can effectively connect human requirements with machine-readable formalisms in OE.

4.1.2 Ontology implementation

The **conceptualization** task in the ontology implementation phase involves defining terms, relationships, and taxonomies. Our analysis identified 8 studies investigating the potential of LLMs in supporting these activities, highlighting it as one of the most prominent research areas in the field.

One key aspect of ontology development is term definition. Bischof et al. (2024) demonstrated that LLMs can substantially reduce the effort required by domain experts by generating context-aware definitions aligned with domain-specific conventions. Beyond definitions, taxonomy discovery and relationship extraction have also been enhanced by LLMs. Goyal et al. (2024) and Babaei Giglou et al. (2023) employed LLMs to support ontology formalization through automated reasoning and relationship identification. Their studies showed that LLMs can detect both hierarchical and non-taxonomic relations between concepts.

Several researchers have proposed integrated frameworks to support ontology conceptualization. For example, Coutinho (2024) developed a system merge text-based languages for ontologies with LLMs to generate new concepts based on contextual information for the unified foundational ontology (UFO) Guizzardi et al. (2015).

Further contributions include Rebboud et al. (2024a), who framed this task as constructing an ontology by generating missing classes and properties. Kholmska et al. (2024) applied LLMs to generate nearly 200 core concepts in the field of active learning, which were hierarchically organized and definitionally refined, demonstrating the potential of LLMs for concept discovery and structuring. Dong et al. (2024) explored zero-shot concept generation, while Toro et al. (2024) introduced techniques for ontology term completion.

Encoding refers to the translation of conceptual models into formal ontology representation languages. Our survey identified 10 studies that investigated how LLMs can support this process. In the context of domain-specific formalization, Doumanas et al. (2024) employed LLMs to develop an OWL ontology for search and rescue missions (SAR). Their evaluation was performed against gold reference ontology in the SAR domain Masa et al. (2022) which has been developed by human experts.

Several studies propose tools for natural language to OWL translation. Mateiu and Groza (2023) created a Protégé plugin[7] that converts natural language sentences into OWL axioms using LLMs. Similarly, Caufield et al. (2024) developed a pipeline that extracts procedural knowledge from websites (e.g., recipes) and have corresponding ontologies. Other works, including Eells et al. (2024) prompted LLMs to generate ontologies for common nouns and assessed the output in terms of syntactic validity and structural completeness. Saeedizade and Blomqvist (2024) investigated the use of LLMs to generate OWL from structured narratives, highlighting the potential of LLMs to assist in transforming textual descriptions into formal ontological representations. Tang et al. (2023) focused on domain-specific knowledge extraction, demonstrating how LLMs can facilitate the construction of ontologies tailored to specific road traffic domain for autonomous vehicles.

Next, da Silva et al. (2024) proposed a method to transform capability descriptions into ontological models using LLMs, streamlining ontology creation from natural language inputs. Fathallah et al. (2024a) developed a pipeline that automates ontology encoding task by instantiating structures with relevant data. Kholmska et al. (2024) proposed an LLM-assisted approach for ontology extension by jointly processing textual and machine-readable data (e.g., OWL/RDF). Additionally, Pisu et al. (2024) explored the use of transformer-based language models for generating research topic ontologies, highlighting the potential of LLMs in taxonomy construction.

LLMs also have been applied to ease the **ontology matching** tasks, which are key to ensure interoperability in diverse knowledge domains. Zamazal (2024) evaluated the effectiveness of LLMs in validating complex mapping candidates, indicating promising results in correspondence validation tasks. Hertling and Paulheim (2023) introduced OLaLa, a system that utilizes LLMs to generate high-precision ontology mappings. Additionally, studies by Norouzi et al. (2023) and He et al. (2023) benchmarked LLM performance in ontology alignment against reference mappings, revealing that modern LLMs can perform comparably to specialized alignment systems. Kholmska et al. (2024) further explored the role of LLMs in generating initial mapping suggestions to support ontology extension, assessing concept coverage and inter-model consistency.

[7] https://protege.stanford.edu/

In the **ontology evaluation** task, LLMs have been employed to assess the quality, consistency, and correctness of ontologies. Tsaneva et al. (2024) utilized ChatGPT-4 to verify ontology restrictions, achieving high accuracy in detecting logical inconsistencies and structural issues. Fathallah et al. (2024a) explored a different take, proposing an evaluation framework that leverages ChatGPT to assist in ontology syntax correction, using parsing errors detected by RDFLib and pitfall descriptions from the OOPS! API Poveda-Villalón et al. (2014), particularly focusing on missing disjointness axioms. This demonstrates that LLMs can not only identify ontology issues but also suggest corrective actions. Similarly, Zhang et al. (2025) introduced OntoChat, a framework for ontology verbalization and validation through prompt-driven unit tests, aiming to make ontology evaluation more accessible to non-expert users. Kholmska et al. (2024) provided a broader evaluation of ontology quality, emphasizing relevance, content consistency, and structural soundness in LLM-supported ontology development.

4.1.3 Ontology publication

The generation of human-readable **documentation** is essential for understanding the definitions and relationships of an ontology. Our analysis identified 4 studies that explore how LLMs can support ontology documentation tasks. Rebboud et al. (2024a) investigated the use of LLMs for generating structured documentation focused on key ontology components such as classes and properties. Their evaluation, which utilized semantic similarity metrics, demonstrated the effectiveness of LLMs in producing accurate and relevant documentation. Fathallah et al. (2024a) addressed the generation of natural language descriptions for ontology entities and properties, improving the comprehensibility and usability of ontologies for both technical and non-technical users. In more specialized applications, Giri et al. (2024) used the T5 model to summarize functional descriptions of Gene ontology terms, while Kholmska et al. (2024) used LLMs to generate comprehensive documentation for extended ontologies, supporting knowledge sharing and reuse.

4.1.4 Ontology maintenance

Among the studies reviewed, only one paper specifically addressed maintenance tasks related to **bug** or detection of **issues**. Kholmska et al. (2024) investigated the potential of LLMs to extract key improvement suggestions, refine task lists, and identify missing concepts from human-evaluated reports(step 8 of their proposed workflow). Their findings suggest that LLM can effectively help maintain ontologies.

4.1.5 Summary

Based on the analysis of 41 studies from 30 papers, LLMs have been applied disproportionately across different phases of ontology development. The implementation phase dominates research attention, with 27 studies focusing on tasks such as conceptualization, encoding, ontology matching and evaluation. Requirements specification represents the second most researched area, with twelve studies exploring how LLMs can generate functional requirements, create competency questions from existing ontologies, and formalize requirements into SPARQL queries. However, later stages of ontology development remain relatively unexplored. Only four studies address ontology publication through documentation generation, while ontology maintenance has received minimal attention with just one study on maintenance.

4.2 RQ2: How do LLMs-based approaches support different ontology development activities?

Following the identification of ontology development tasks supported by LLMs in Section 4.1, we now explore the internal workings of how LLMs contribute to these tasks. This includes analyzing their

functional roles (as ontology engineers or domain experts, etc.), model choices (from GPT series or other open-source tools), input and output types utilized by LLMs, and whether the studies collaborate with humans in the LLM-based activities. Table 1 displays the inputs and outputs associated with each ontology development activity. For a more detailed breakdown, including specific model names, functional roles, and human collaboration status, we refer the reader to Table 2 in the Appendix.

4.2.1 RQ2.a: What is the role of LLMs models in OE activities?

From the analysis of the selected studies, LLMs are found to play several key collaborative roles in ontology-related activities. These roles complement and, in some cases, replicate the tasks traditionally performed by human experts in knowledge engineering. We categorize these roles into four primary types:

1. **Ontology Engineer**: LLMs often function as Ontology Engineers, supporting the design, development, and maintenance of ontologies throughout the entire ontology development lifecycle. More precisely, LMs are used to (a) parse unstructured domain texts and generate structured requirement specifications thus facilitating automated requirement elicitation Alharbi et al. (2024a); Ciroku et al. (2024a); (b) to transform competency questions into structured queries (e.g., SPARQL) Rebboud et al. (2024a); Tufek et al. (2024); (c) to discover axioms, particularly to identify hierarchical relationships between concept pairs, during the conceptualization activity Goyal et al. (2024); Babaei Giglou et al. (2023); (d) translate unstructured or semi-structured texts into OWL code Doumanas et al. (2024); Eells et al. (2024); Saeedizade and Blomqvist (2024); Tang et al. (2023); (e) the entire ontology lifecycle, from conceptualization to documentation Kholmska et al. (2024) or end-to-end under the NeOn-GPT methodology Fathallah et al. (2024a).

2. **Domain Experts**: LLMs also take on the role of domain experts by assisting in knowledge extraction, term definition, and ontology content validation. LLMs have been applied during tasks requiring domain-specific understanding, for example, (a) to generate domain-relevant concepts Dong et al. (2024); to produce terms definitions Bischof et al. (2024); (c) to generate ontology documentation Rebboud et al. (2024a); Giri et al. (2024) and to summarize functional descriptions Consortium (2006). They have been also applied in evaluation tasks that not only required technical knowledge but also domain specific background to evaluate consistency and correctness Tsaneva et al. (2024); Fathallah et al. (2024a).

3. **Human Evaluator**: In some cases, LLMs have been placed as human evaluators, for example, to verify ontology axioms and assess their logical soundness Tsaneva et al. (2024).

4.2.2 RQ2.b: What types of LLMs are used in OE activities?

The LLMs employed in OE span a range of architectures and capacities. Based on our analysis, these models can be grouped into four major categories, each playing distinct roles in the OE lifecycle.

- **GPT series (GPT-3.5, GPT-4, GPT-4 Turbo/4o):** The GPT series is among the most widely used for tasks involving CQ reverse engineering, encoding, and evaluation, owing to their strong capabilities in natural language understanding and generation Rebboud et al. (2024b); Tufek et al. (2024); Fathallah et al. (2024a). In particular, GPT-4 Turbo/4o has been leveraged for more complex tasks requiring multi-formalism reasoning, such as verifying axioms across heterogeneous logical representations Zamazal (2024).

- **Open-Source large language models (LLaMA, Mistral, Claude, etc.):** Open source LLMs such as LLaMA Touvron et al. (2023a), Mistral[8] and Claude[9] are also used mainly in ontology development tasks, including functional requirement writing, conceptualization, encoding, etc.

  Hertling and Paulheim (2023) fine-tuned LLaMA for ontology matching and reuse, aligning anatomy ontologies in the OAEI benchmark. Goyal et al. (2024) leveraged LLaMA3 and Mistral to detect hierarchical relations in GeoNames and Schema.org. Saeedizade and Blomqvist (2024) combined LLaMA-generated outputs with expert feedback to iteratively refine a SAR ontology. da Silva et al. (2024) demonstrated that Claude 3 and Gemini Pro can effectively convert natural language descriptions into OWL axioms, supporting the ontology encoding process. Additionally, LLaMA and PaLM were integrated into the NeOn-GPTframework proposed by Fathallah et al. (2024a), supporting multiple stages of ontology development, including functional requirements, encoding, evaluation, and documentation.

- **Lightweight Instruction-Tuned Models (e.g., Mistral-7B, Falcon-7B-Instruct, etc.):** Lightweight instruction-tuned models have been applied in OE tasks, as demonstrated in two recent studies. Alharbi et al. (2024b) employed models such as LLaMA-2-70B, Mistral 7B Chaplot (2023), and Flan-T5-XL to generate CQs by embedding RDF triples into prompt templates enriched with varying levels of contextual information. The resulting CQs were then filtered to produce a final set of relevant, non-redundant questions. Saeedizade and Blomqvist (2024) further explored the use of lightweight open-source models including LLaMA-7B, LLaMA-13B, LLaMA-2-70B, Alpaca, Falcon-7B, and Falcon-7B-Instruct—for ontology encoding. Their study demonstrated the capability of these models to process narrative ontology descriptions and associated CQs for automated ontology creation, in comparison with models such as GPT-3.5, GPT-4, and Bard.

- **Transformer-Based Architectures (T5, BERT):** Beyond large-scale LLMs, pre-trained transformer models, such as T5 and BERT, are powerful in supporting sentence encoding, classification, and structured generation. Ciroku et al. (2024b) used T5 and SBERT within the RevOnt framework to automatically extract competency questions from knowledge graphs. Giri et al. (2024) applied T5 in the GO2Sum system to generate human-readable functional descriptions of Gene ontology terms, supporting ontology documentation and publication. Furthermore, Pisu et al. (2024) proposed the use of SciBERT for the generation of taxonomy of research publication topics, with the objective of integrating domain-adapted language models into ontology encoding and KG construction workflows.

4.2.3 RQ2.c: What are the inputs given to LLMs? and RQ2.d: What are the outputs from the LLMs?

To better analyze the use of LLMs during the OE lifecycle, we contextualize the input given to the LLMs in relation to the expected output and the specific task at hand. For this reason, in this section we report the results obtained for RQ2.c and RQ2.d and group the results according to RQ1 which acts as the backbone for this survey and a natural way of grouping, as the OE activities are driven by the type of their expected output.

[8] https://mistral.ai/
[9] https://www.anthropic.com/

- During the **ontology requirement specification** phase there are common patterns depending on the activity at hand. More precisely: (a) taking as input natural language text to write functional requirements either in the shape of CQs (Antia and Keet (2023)) or natural language affirmative statements (Fathallah et al. (2024a)); (b) transforming structured inputs (ontologies, triples or KGs) to write CQs through reverse engineering (Rebboud et al. (2024a); Alharbi et al. (2024a); Ciroku et al. (2024b); Rebboud et al. (2024b); Alharbi et al. (2024b)); and (c) taking ontologies and natural language (including CQs) to generate queries as part of the requirement formalization activity.

- For the **ontology implementation** phase, there are common patterns for activities with clear output formats such as ontology encoding and ontology matching. However, approaches addressing less restricted activities, such as ontology conceptualization or evaluation, present higher variability. More precisely:

  – While all approaches take natural language text as input in different formats, as is typically the case for OE projects, the **ontology conceptualization** activity leads to various types of outputs. Some approaches generate machine-readable representations, such as ontologies in OWL (Rebboud et al. (2024a)) or structured schemas in JSON or YAML Toro et al. (2024). Others produce concepts or terms intended for ontology integration Bischof et al. (2024); Kholmska et al. (2024); Babaei Giglou et al. (2023). Also, some approaches generate natural language descriptions (Dong et al. (2024)), or binary decisions to validate semantic relations or classify term types (Goyal et al. (2024); Babaei Giglou et al. (2023)). A special classification task is presented by Pisu et al. (2024), to predict semantic relations (e.g., supertopic, subtopic, same-as, other) between topic pairs extracted from an existing ontology.

  – For the **ontology encoding** activity, most analyzed approaches (Doumanas et al. (2024); Fathallah et al. (2024a); Caufield et al. (2024); Eells et al. (2024); Mateiu and Groza (2023); Tang et al. (2023); da Silva et al. (2024)) take natural language descriptions as input to generate ontology artifacts in OWL, RDF, or related formats. An exception is Saeedizade and Blomqvist (2024), which uses competency CQs as input to guide ontology generation in alignment with user information needs. Regarding outputs, most approaches produce complete ontology code, with exceptions like Mateiu and Groza (2023) which focuses specifically on generating OWL axioms. In Eells et al. (2024), the LLM is prompted with a single noun (e.g., “air,” “book”) and returns a mix of natural language text and RDF ontology content.

  – To address **ontology matching**, some of the analyzed works take natural language inputs to produce binary decisions indicating semantic alignment. For example, Zamazal (2024) uses LLMs to classify verbalized complex correspondence candidates as (probably) positive or negative, while He et al. (2023) evaluates the equivalence of concept pairs based on their names and hierarchical contexts, outputting a "Yes" or "No" response. Other approaches directly generate ontology mappings. Norouzi et al. (2023) takes structured representations of two ontologies (in the form of subject–predicate–object triples), and outputs a set of proposed alignments between classes or properties. Kholmska et al. (2024) approaches ontology reuse through a multi-step process: Step 4 employs LLMs to extract key features—such as purpose, reused elements, and formats from existing ontologies to support reuse decisions; Step 6 involves using LLMs to map new concepts to the previously

identified ontologies by analyzing their definitions, relationships, and properties. The output consists of explicit mappings expressed as `owl:sameAs`, `owl:equivalentClass`, and `owl:equivalentProperty` statements. Similarly, Hertling and Paulheim (2023) combines textual and structural information to generate formal ontology alignments.

– For **ontology evaluation** some approaches take natural language text as input (Tsaneva et al. (2024); Fathallah et al. (2024a)), which can include evaluation reports (Fathallah et al. (2024a); Kholmska et al. (2024)), while others also utilize structured ontology-related information or ontologies (Kholmska et al. (2024); Zhang et al. (2025)). The ontology evaluation activity results in various types of output. Some approaches generate machine-readable corrections or modifications, such as class value replacements or the addition of disjointness axioms Fathallah et al. (2024a). Others produce natural language assessments regarding ontology relevance, structural completeness, and alignment with standard frameworks such as CRISP-DM Kholmska et al. (2024). Another line of work focuses on classifying and verifying individual axioms as correct or defective, optionally specifying the type of modeling defect Tsaneva et al. (2024). Alternatively, one study outputs binary decisions such as Yes/No judgments to validate the coverage of CQs based on the ontology content Zhang et al. (2025)

- To address **ontology documentation** activity, all analyzed approaches focus on generating human-readable documentation. They take ontologies or terms as input (Rebboud et al. (2024a); Giri et al. (2024)), and optionally incorporate additional natural language text sources (Kholmska et al. (2024); Fathallah et al. (2024a)). Specifically, Rebboud et al. (2024a) emphasizes the production of readable summaries highlighting key classes and properties. Giri et al. (2024) generates concise summaries from Gene Ontology terms. Finally, Kholmska et al. (2024) leverages LLMs to assist in the writing of technical reports.

- The only work explicitly addressing **ontology maintenance** is Kholmska et al. (2024), where LLMs are used to support iterative refinement. In Step 8 of their workflow, domain expert feedback and validation reports serve as input. These documents are uploaded to an LLM interface, where the model reviews the content, extracts improvement suggestions, and generates refined task lists. The output is human-readable text highlighting missing concepts, potential relationship issues, and areas requiring adjustment within the ontology. While the ontology itself is not used as direct input, its structure is implicitly referenced through the content of the validation reports.

4.2.4 RQ2.e: What is the role of humans in OE LLM-assisted activities?

Although many recent studies automate ontology development with LLMs, only 4 studies explicitly involve human participants, typically domain experts or ontology engineers, to support tasks requiring judgment, contextual understanding, and refinement.

Doumanas et al. (2024) highlight the pivotal role of domain experts during the ontology encoding phase. Experts were responsible for evaluating both existing ontologies and LLM-generated semantic content, ultimately steering the model toward the creation of a new ontology tailored for SAR operations. Similarly, Kholmska et al. (2024) describe the involvement of domain experts and end-users during ontology maintenance and bug resolution. Their iterative feedback on errors and inconsistencies was critical for refining the ontology structure and enhancing overall quality. In the context of ontology evaluation, Zhang et al. (2025) demonstrate how ontology engineers curated user stories that were manually authored or

derived from earlier development stages to support meaningful CQ extraction, emphasizing the necessity of human input in linking technical outputs to real-world use cases. Finally, Alharbi et al. (2024a) report interviewing human experts and ontology engineers to capture design intentions. These insights were then used to generate contextually accurate CQs, particularly in support of functional specification and requirements engineering.

4.2.5 Summary

Based on the results presented in Section 4.2.1, LLMs have been found to assume several key roles within ontology engineering workflows. As **Ontology Engineers**, they automate the generation and refinement of ontologies from unstructured text and formalize competency CQs into executable queries. As **Domain Experts**, they assist in knowledge extraction, term definition, and domain-specific validation. As **Human Evaluators**, LLMs like ChatGPT demonstrate strong performance in verifying ontology axioms and detecting logical inconsistencies. These roles underline the versatility of LLMs, with research focusing primarily on the ontology engineer and domain expert roles, while the evaluator role remains underexplored but promising for advancing ontology validation.

According to Section 4.2.2, LLMs applied in Ontology OE can be categorized into four main types: GPT series, Open-Source Large Models, Lightweight Instruction-Tuned Models, and Transformer-Based Architectures. GPT models (e.g., GPT-4) are the most prevalent, supporting tasks such as competency question generation, ontology encoding, and evaluation. Open-source models like LLaMA and Mistral are widely applied in ontology matching and conceptualization. Lightweight instruction-tuned models (e.g., Mistral-7B, Falcon-7B-Instruct) offer efficient solutions for CQ reverse engineering and ontology creation. Transformer-based models such as T5 and BERT mainly assist in knowledge extraction and ontology documentation. Overall, GPT-based models currently dominate complex and reasoning-intensive OE tasks, while open-source and lightweight models are increasingly favored for their adaptability and efficiency. Transformer-based architectures remain valuable in structured knowledge extraction and publication-related activities, reflecting a diverse ecosystem of LLM applications across the ontology development lifecycle.

From Section 4.2.3, LLMs are shown to process diverse inputs and generate varied outputs across different OE phases. Inputs range from unstructured texts, such as domain-specific corpora, competency question templates, and requirements, to semi-structured and structured data, including ontology files, RDF triples, taxonomy tuples, and knowledge graph subgraphs. In early phases like requirements specification and conceptualization, natural language inputs dominate, while structured inputs become more prevalent in later tasks such as encoding, matching, and evaluation, reflecting increasing demands for formality and precision. LLM outputs align with the task and input type. In requirements specification and CQ reverse engineering, outputs are typically structured texts or executable queries (e.g., SPARQL). In conceptualization and encoding, LLMs produce formal artifacts like OWL axioms, complete ontology files, or structured formats such as JSON and YAML. Matching and evaluation tasks yield binary decisions, mappings, or validated axioms, while publication and documentation focus on human-readable descriptions and summaries for knowledge dissemination.

Finally, as reported in Section 4.2.4, although LLMs significantly reduce manual workload across various ontology development tasks, human expertise remains essential for critically assessing both existing ontologies and LLM-generated outputs to ensure semantic alignment with domain requirements.

4.3 RQ3: How is the performance of LLMs in ontology development evaluated?

In this section, we analyze the experimental support provided in the reviewed studies to validate their proposed frameworks and methodologies. Specifically, we examine whether these studies include experiments and whether they are open-source, as transparency is essential for reproducibility and independent validation. We also investigate the datasets used in these studies to determine if a common benchmark was used across different studies. Most importantly, we assess the performance of LLMs in ontology development, focusing on the evaluation methods (quantitative, qualitative, or hybrid) and the specific metrics used, such as F1, BLEU, or others. These details allow us to thoroughly assess the reported performance results from these papers and evaluate the effectiveness of LLMs in addressing various ontology engineering challenges. Table 3 in Appendix 7 compiles and summarizes all information on the availability of experiments, datasets used, evaluation types, and evaluation metrics applied across reviewed studies.

4.3.1 RQ3.a Does an experiment exist?

Of the 41 reviewed studies, eleven studies were conducted without experiments. Four out of 41 provide source experiments Fathallah et al. (2024a), but only tests LLMs without baselines or comparisons to demonstrate performance. The remaining 27 out of 41 studies include experiments with evaluation metrics and comparative analysis in their tasks. However, three out of 27 reporting experiments in their publications lack access links Tsaneva et al. (2024); Alharbi et al. (2024b); Norouzi et al. (2023).

It should be noted that Fathallah et al. (2024a) and Kholmska et al. (2024) address multiple tasks, ranging from requirements specification to addressing bug issues. Therefore, these approaches appear multiple times in our analysis.

4.3.2 RQ3.b What data sets are used in the evaluations?

Although not all studies include full experiments, many of them still explicitly mention the datasets used in their work. An exception is the study by Bischof et al. Bischof et al. (2024), which does not specify any dataset names. Instead, it refers to terms or concepts without providing detailed information about the data sources involved. Through the other 41 studies, all explicitly describe their dataset types and provide publicly accessible datasets through platforms such as GitHub, Zenodo, or some official ontology database (e.g., GO, OntoDM Ontology). From these 41 studies, 38 use ontology related files (OWL, RDF, etc.) as their primary dataset type.

Drilling into the detailed datasets used, Alharbi et al. Alharbi et al. (2024b) selected four ontologies along with their associated CQ datasets to investigate CQ creation. Three of these ontologies: Video Game (entertainment), Dem@care (healthcare), and VICINITY Core (Internet of Things) were obtained from the CORAL Fernández-Izquierdo et al. (2019) repository, a comprehensive source for CQs. The fourth ontology, African Wildlife (ecology) Keet (2019), was included to ensure diversity in both domain coverage and CQ styles.

Meanwhile, Dong et al. (2024) applied the MM-S14-Disease and MM-S14-CPP datasets Dong et al. (2023), both from the biomedical domain, to evaluate LLM performance in ontology mapping. After encoding the ontologies into OWL using syntax-aware concepts derived from textual descriptions, they leveraged version differences in SNOMED CT Donnelly et al. (2006), a clinical terminology system, to define new concepts and construct ground-truth placement edges. Similarly, Kholmska et al. (2024) used the OntoDM suite Panov et al. (2008) and IOF Core Drobnjakovic et al. (2022), both rooted in the

industrial engineering domain, due to their maturity, comprehensive documentation, and validation within real-world manufacturing settings.

Ciroku et al. (2024a) introduced the first implementation of RevOnt, which leverages the Web Data Visualizer Knowledge Graph (WDV) Amaral et al. (2022) constructed from Wikidata Vrandečić and Krötzsch (2014), a collaborative knowledge. WDV comprises 7.6K unique RDF triples and includes manually annotated competency questions, providing explicit subject–predicate–object relationships that serve as ground truth for CQ derivation. This resource enables the quantitative evaluation of data verbalization models (e.g., via BLEU score), comparing LLM-generated questions to human-authored ones. Tsaneva et al. (2024) used food domain Pizza Ontology in Protégé to benchmark LLM-driven defect detection in OWL axioms. Giri et al. (2024) focused on the summarization of protein functions in the bioinformatics domain, evaluating the generated outputs against GO Consortium (2006), a fundamental resource in molecular biology. Similarly, Toro et al. (2024) evaluated the quality of LLM-generated definition generation for biomedical Cell ontology Diehl et al. (2016) using BERTScore, supplemented with manual expert review to ensure semantic validity.

We also observed that several studies share common experimental ontologies, enabling standardized evaluation and comparative analysis. For ontology matching tasks, studies such as Zamazal (2024), Hertling and Paulheim (2023), and Norouzi et al. (2023) utilized datasets from the OAEI 2022 benchmark tracks, which provide both ontologies and KGs across diverse domains. Similarly, Babaei Giglou et al. (2023) and Goyal et al. (2024) adopted benchmark ontologies from the LLMs4OL Challenge, designed to assess LLMs across various ontology learning tasks. This challenge spans multiple domains, including WordNet (lexical) Miller (1995), GeoNames (geospatial) Volz et al. (2007), UMLS Bodenreider (2004) and SNOMED CT (biomedical) Donnelly et al. (2006), and schema.org (web) Guha et al. (2016) [10] ontologies. These shared benchmarks facilitate consistent evaluation of LLM-based methods in structured knowledge engineering

Furthermore, several data sets have been reused in studies to enable consistent evaluation in tasks and models. For example, Fathallah et al. (2024a) used the Wine Ontology as a gold standard in their NeOn-GPT pipeline, covering tasks such as requirements writing, OWL encoding, publication, and documentation. Rebboud et al. (2024a) and Rebboud et al. (2024b) evaluated LLM-generated outputs using a consistent set of ontologies: DOREMUS Achichi et al. (2018), Polifonia de Berardinis et al. (2023), Dem@Care Karakostas et al. (2016), Odeuropa Lisena et al. (2022), NORIA-O Tailhardat et al. (2024), and FIBO Bennett (2013) across multiple tasks including CQ reverse engineering, conceptualization, and ontology documentation.

In addition to ontology files, several studies have explored the use of unstructured datasets and natural language texts as experimental inputs. Mateiu and Groza (2023) used 150 unstructured descriptions of ontological elements to evaluate a Protégé plugin that translates natural language sentences into OWL axioms. In a different setting, Antia and Keet (2023) employed COVID-19 scientific papers as input to an automated CQ reverse engineering pipeline, aiming to extract meaningful queries for ontology validation. Eells et al. (2024) focused on ontology construction, using 101 high-frequency nouns from the Corpus of Contemporary American English (COCA) Davies (2010) as prompts. These nouns that cover general concepts were used to guide LLMs in generating ontological structures, which were then evaluated for semantic coherence and alignment with human common sense knowledge. To support further exploration

[10] https://schema.org

of datasets used in LLM-based ontology engineering tasks, we provide Table 5 in Section 7. The table lists acronym and full name of datasets, official or commonly used access link, and its associated domain, helping readers identify suitable datasets for specific domain applications.

4.3.3 RQ3.c: What evaluation methods are used?

Next, we examine how performance is evaluated, i.e., whether it is based on comparison against a reference standard or by applying specific scoring metrics. We also look at the type of evaluation methods used, distinguishing between quantitative approaches and qualitative approaches. Additionally, we explore the evaluation metrics applied in these studies, such as F1, BLEU score, or human evaluation criteria. Finally, we assess whether human involvement is included in the evaluation process, through expert reviews or manual selection.

Out of the 41 reviewed studies, we found that 9 studies do not conduct any evaluation. This includes works that either lack experimental implementation entirely Tang et al. (2023); Mateiu and Groza (2023), as noted in Section 4.3.2, or only present basic demonstrations without comparative baselines or metric based analysis Fathallah et al. (2024a). Additionally, two studies Kholmska et al. (2024); Bischof et al. (2024) describe evaluation strategies but do not report actual results, reflecting only a conceptual treatment of evaluation. Moreover, some studies such as Kholmska et al. (2024) involve multi-step pipelines in which only selected tasks are evaluated, often excluding pre-processing or intermediate components. Here we focus on the papers that include an evaluation. The remaining studies evaluate the development of LLM-driven ontology using based on three main approaches, as described below.

- **Quantitative Evaluation Approach**
  Most studies adopt quantitative methods, using automated metrics to assess LLM performance:
  - **Performance-based evaluation:** Metrics such as precision, recall, and F1-score are widely used, alongside specialized metrics like inter-model consistency or error rate reduction, particularly in tasks like ontology matching and conceptualization. For example, Hertling and Paulheim (2023) evaluated ontology matching results using precision, recall, and the F1 score, compared to the OAEI datasets. Similarly, Goyal et al. (2024) and Babaei Giglou et al. (2023) apply the F1 score to measure the accuracy of LLM-generated outputs in ontology conceptualization tasks, as part of the LLMs4OL challenge. Alharbi et al. (2024b) and Kholmska et al. (2024) report task-specific metrics such as intermodel consistency, error rate reduction, and concept coverage to assess the quality of generated ontologies. Dong et al. (2024) evaluate hierarchical relation predictions using the Insertion Rate at top k (InR@k), which reflects how accurately new concepts are inserted into a taxonomy. Tufek et al. (2024) measure the accuracy of the exact match for the generation of SPARQL queries by comparing the outputs with predefined targets.
  - **Similarity-based evaluation:** Some studies apply semantic similarity measures, such as SentenceBERT cosine similarity, to compare LLM-generated outputs against reference texts, reducing the need for manual comparisons. Rebboud et al. (2024b) use SentenceBERT cosine similarity to evaluate the semantic relationship between LLM-generated competency questions and expert references. In a related setting, Rebboud et al. (2024a) apply cosine similarity to compare generated ontology documentation with expert definitions, supporting an efficient and consistent quality assessment.

  - **Ground-truth-based evaluation:** Structural fidelity is evaluated through metrics like tree edit distance (for SPARQL queries) Rebboud et al. (2024a) or BLEU score (for generated CQs) Ciroku et al. (2024a), ensuring alignment with gold standard datasets. While BLEU focuses on surface level lexical similarity, it remains a valuable metric of textual fidelity in structured natural language generation tasks, particularly in the context of CQ reverse engineering .

- **Qualitative Evaluation Approach**
  A smaller number of studies employ only human based evaluation. Domain experts assess LLM outputs based on semantic precision, conceptual correctness, and domain relevance, providing critical insights beyond automated metrics. Zhang et al. (2025) utilizes a qualitative assessment approach through expert-driven questionnaires, where ontology engineers and domain experts provide nuanced feedback. Bischof et al. (2024) incorporates a rigorous qualitative evaluation that relies on experts in their work, in which specialized experts meticulously assess the definitions generated by LLMs for semantic precision, conceptual precision, and domain-specific correctness.

- **Hybrid Evaluation Approach**
  Several studies integrate both quantitative and qualitative evaluations. They combine metric-based assessments with expert reviews to validate both the structural quality and practical usability of the LLM outputs, enhancing evaluation robustness. da Silva et al. (2024) combine SHACL-based syntax checks with expert review to ensure logical consistency and eliminate redundancy in generated ontologies. Giri et al. (2024) integrate human evaluation to validate the embedding-based confidence scores used for assessing LLM-generated biomedical summaries. The study examines the correlation between automated and expert ratings, particularly when high embedding scores are observed. Coutinho (2024) adopt a hybrid evaluation approach, combining quantitative measures (e.g., task completion time, model quality metrics) with qualitative insights from expert interviews and user satisfaction assessments. This strategy improves inter-model consistency and enhances overall usability by balancing automation with human feedback. Alharbi et al. (2024a) also implement both quantitative and qualitative evaluation. On the quantitative side, they compare the number and distribution of generated CQs against existing ones, using metrics such as mean questions per triple, precision, recall, and F1-score. Qualitatively, they conduct expert interviews with ontology developers to assess the intent and relevance of generated CQs, further involving ontology editors to rate predicted versus curated definitions. Finally, Tsaneva et al. (2024) examine the use of ChatGPT-4 for verifying ontology restrictions. The study compares the performance of LLM-driven evaluations against human expert assessments to determine the feasibility and reliability of automated verification.

4.3.4 RQ3.d: What are the performance results from the evaluation?

As reported in previous sections, the reviewed works use different input datasets and metrics, and hence are not directly comparable. However, here we discuss the overall reported results, grouped by activity, to obtain a qualitative overview of the state of the art.

Among the requirements specification phase, LLMs are reported to be effective in producing CQs aligned with original ontology design intentions, achieving high recall across various benchmarks Rebboud et al. (2024b); Antia and Keet (2023); Alharbi et al. (2024a). Proprietary models consistently outperformed open-source ones, while the latter showed greater variance in performance (recall ranging from 0.58 to

1.00), largely due to differences in training data and architecture Rebboud et al. (2024a). Lower temperature settings were found to reduce hallucinations without compromising accuracy Rebboud et al. (2024a).

Notably, the RevOnt framework Ciroku et al. (2024a) achieved strong performance, with a median BLEU score of 0.41 in verbalization and 0.30 in question generation. Over 75% of its outputs were rated as good to high quality, particularly excelling in object-centric questions. Rebboud et al. (2024b) focused on automated CQ reverse engineering as the primary task. Specifically, *Zephyr* β and *UNA* achieved high precision when evaluated on their ability to generate relevant CQs for RDF-based ontologies such as DOREMUS and Odeuropa. Furthermore, the *AgOCQs* framework Antia and Keet (2023) demonstrated strong performance in generating CQs aligned with ontology design expectations. In a manual evaluation by domain experts, over 80% of the LLM-generated CQs were rated as both grammatically correct and semantically relevant, indicating the effectiveness in model to produce high quality queries.

In a complementary line of work, the RETROFIT-CQ pipeline Alharbi et al. (2024a) focused on adapting CQs to existing ontologies by generating questions from RDF triples. The evaluation showed that more than 75% of the generated CQs were directly executable as SPARQL queries without requiring manual revision, demonstrating the development of high structural ontology compatibility. In the requirement formalization task, LLMs demonstrated strong performance in translating natural language into SPARQL queries. Tufek et al. (2024) reported F1 scores ranging from 88% to 96%, with prompt template optimization significantly enhancing output quality. Execution modality also mattered: using a web interface yielded 100% F1, outperforming API-based execution (93%).

In the ontology implementation phase, particularly in conceptualization, GPT-4o demonstrated strong zero-shot performance in the LLMs4OL challenge tasks, achieving an F1 of 72.78% and winning six subtasks Goyal et al. (2024); Babaei Giglou et al. (2023). Fine-tuning of the Flan-T5 models led to substantial improvements, 25% in Task A and 45% on WordNet-related tasks. In domain-specific ontology construction, SciBERT achieved 91.29% F1 and over 91% accuracy by supporting term typing and taxonomy discovery Pisu et al. (2024). For hierarchical concept placement, models enhanced with explainability-driven instruction tuning, such as LLaMA-2-7B, outperformed larger general-purpose LLMs Dong et al. (2024).

During encoding and implementation, prompt engineering and iteration were reported to produce mixed results. In the SPIRES framework, GPT-3.5-turbo enabled perfect entity alignment, but for zero-shot chemical-disease relation tasks, SPIRES achieved 43.8 F1 Caufield et al. (2024). Claude and similar fine-tuned models showed superior performance in constraint generation, outperforming the baseline GPT in capability ontology generation da Silva et al. (2024). In SAR use cases, GPT-4 with Chain-of-Thought prompting produced *reusable* OWL ontologies Doumanas et al. (2024), i.e., ontologies that exhibit semantic consistency, modular design, and generalizability across multiple domains.

In ontology matching and reuse, GPT-4o correctly validated complex alignments with 100% accuracy in rejecting false correspondences Zamazal (2024). The OLaLa study showed F1 score improvements with LLaMA 2 (70B), optimized for efficiency Hertling and Paulheim (2023). Flan-T5-XXL also performed best overall in alignment tasks across benchmarks He et al. (2023), while conversational prompting approaches reported balanced recall and precision, benefiting from expert feedback Norouzi et al. (2023).

In ontology evaluation, ChatGPT-4 verified axioms with 92.2% accuracy, improving to 96.7% via ensemble aggregation Tsaneva et al. (2024). OntoChat received 87.5% positive ratings from experts for clustering competency questions Zhang et al. (2025). DRAGON-AI demonstrated high precision but moderate recall, iteratively improving with user input Toro et al. (2024). In a controlled educational

setting, LLMs using CQ-by-CQ prompting approximated student-level ontology quality Saeedizade and Blomqvist (2024).

Finally, in the bug issue task, GO2Sum Giri et al. (2024) shown strong performance in summarizing Gene ontology annotations. It produced readable and semantically coherent descriptions even for low-confidence predictions. These scores reflect the percentage of summaries judged as helpful for understanding low-scoring Gene ontology predictions, indicating the effectiveness of LLMs in supporting ontology debugging and interpretation.

4.3.5 Summary

From our findings in Section 4.3.1, 27 out of the 41 reviewed studies conducted full experiments, including the use of evaluation metrics and comparative analysis. This emphasizes the growing importance of open-source and highlights a clear trend toward empirical validation in LLM-based OE research.

From Section 4.3.2, it is evident that almost all reviewed studies explicitly specify the datasets used in their evaluations. Across these studies, ontology files (e.g., OWL, RDF) are widely adopted as the primary evaluation resources. These structured datasets often serve as gold standards, being developed by domain experts with rich resources such as user stories, CQs, documentation, and queries to ensure high quality and reliability, making them well suited for evaluating tasks like CQ reverse engineering, ontology encoding, and ontology evaluation.

Several well established benchmark ontologies such as the Stanford Wine Ontology, and domain specific ontologies like Polifonia and FIBO have been reused across multiple studies, supporting consistent evaluation across different tasks. In addition, certain benchmarks have been designed specifically for particular tasks; for instance, the OAEI 2022 tracks target ontology matching, while the LLMs4OL Challenge provides customized datasets for ontology conceptualization and learning tasks. These resources facilitate the quantitative evaluation of LLM-generated ontologies across key metrics such as alignment accuracy, domain coverage, and redundancy reduction.

As for the valuation methods reported in Section 4.3.3, we find a preference of performance-based evaluation methods, followed by similarity-based evaluations and comparisons against ground-truth data. So far no standard set of metrics has been established for any particular task. Hybrid evaluation approaches, while less common, attempt to include expert feedback together with quantitative results (e.g., through interviews with experts).

Next, according to the performance results reported in Section 4.3.4, many studies demonstrate positive outcomes when applying LLMs to specific OE phases, such as encoding or CQ reverse engineering. However, most evaluations focus on the effectiveness of the overall pipeline, without isolating the specific contribution of LLMs at each stage. For instance, Alharbi et al. (2024a) applied LLMs for CQ reverse engineering but assessed performance in the context of ontology implementation. Similarly, da Silva et al. (2024) and Fathallah et al. (2024a) used LLMs for encoding tasks, yet evaluated success across multiple downstream activities.

Moreover, even when studies target the same OE tasks or participate in shared challenge settings, fair performance comparison remains difficult. For example, Zamazal (2024), Hertling and Paulheim (2023), and Norouzi et al. (2023) all conducted ontology matching using datasets from the OAEI 2022 benchmark, but differences in LLM selection, methodological design, and evaluation metrics hinder meaningful comparison. Likewise, although Babaei Giglou et al. (2023) and Goyal et al. (2024) participated in the same LLMs4OL Challenge, they employed different models and focused on distinct subtasks.

Therefore, performance results are inherently shaped by the choice of LLMs, integration strategies, and evaluation criteria, which limits comparability across studies. Consequently, the evidence supports that LLMs perform well within the specific context of each individual study, rather than enabling generalized claims about the overall superiority of a particular approach.

4.4 RQ4: What are the main application domains where LLMs have been applied in the development of ontology?

In this section, we examine the domain-specific applications of LLMs in OE. *Healthcare and life sciences* represent one of the most extensively explored areas. LLMs have been applied to validate ontological constraints in major biomedical terminologies such as SNOMED CT and UMLS Tsaneva et al. (2024), and to assist in developing domain-specific ontologies like DemCare for dementia care Rebboud et al. (2024a,b). Furthermore, they support biomedical knowledge enrichment tasks in widely adopted resources such as the GO, MONDO, and the Cell Ontology, either by generating functional summaries Giri et al. (2024) or extending axioms and class definitions Caufield et al. (2024); Toro et al. (2024). *Cultural heritage industries* also benefit from LLMs. Ontologies such as DOREMUS, Polifonia, and Odeuropa are enhanced for music and olfactory heritage representation Rebboud et al. (2024a,b); Zhang et al. (2025). In the *finance* domain, LLMs are used for automated CQ reverse engineering and benchmarking of ontologies such as the Financial Industry Business Ontology (FIBO) Rebboud et al. (2024a,b), thus contributing to a more systematic knowledge organization in regulatory and investment contexts. Within *emergency and safety* domain, LLMs have been utilized to construct SAR ontologies based on related knowledge, including environmental conditions, hazard classification, and resource planning through structured prompting strategies Doumanas et al. (2024). In the autonomous systems and smart technologies domain, LLMs have been used to model traffic scenarios in autonomous driving ontologies Tang et al. (2023) and to define concepts for smart building systems Bischof et al. (2024), allowing automation and validation processes. For *academic and research domains*, LLMs help structure and classify research topics, as seen in the Computer Science Ontology (CSO) Pisu et al. (2024), offering scalable solutions for scientific knowledge organization and retrieval. In the *food* field LLMs support the enrichment of ontologies like FoodOn by extracting structured data from recipe texts Caufield et al. (2024), aiding in the classification of ingredients, preparation methods, and nutritional profiles.

4.4.1 Summary

Overall, these applications highlight the versatility of LLMs across diverse ontology-driven domains, as summarized in Table 4 in Section 7. While life sciences, healthcare, and cultural heritage are currently the most active areas, emerging fields such as autonomous systems, disaster management, and regulatory compliance are increasingly adopting LLM-based ontology engineering solutions. This cross-domain applicability demonstrates the potential of LLMs to enhance both the scalability and adaptability of ontology development workflows.

5 Discussion

Below, we explore the implications of our findings in relation to our RQs, highlighting the challenges and opportunities they present.

5.1 Supporting ontology development activities with LLMs

Among the studies reviewed, LLMs have been integrated into multiple key activities throughout the ontology development lifecycle. Our analysis shows that activities related to ontology implementation, particularly conceptualization and encoding, have received greater research attention compared to tasks such as requirements specification, evaluation, and maintenance during the ontology development process.

LLMs demonstrate significant advantages in the early stages of OE, particularly in foundational tasks such as domain conceptualization and hierarchical knowledge structuring. By leveraging their powerful natural language understanding and generative reasoning capabilities, LLMs have shown potential to automatically extract domain-specific concepts, induce class hierarchies, and identify relational patterns from unstructured text. Empirical studies have shown that the candidate concepts and taxonomies generated by LLMs can approximate the quality of human-annotated gold standards in terms of scalability and semantic coherence, often aligning closely with expert-curated ontologies Caufield et al. (2024); da Silva et al. (2024); Doumanas et al. (2024).This enables the rapid generation of structured knowledge representations, meeting common ontology development needs.

Nevertheless, despite these advances, notable limitations persist. The application of LLMs across the ontology lifecycle remains uneven, with tasks such as documentation, evaluation and maintenance receiving significantly less attention. Some of these phases demand deep domain expertise, dynamic contextual reasoning, and strict logical consistency capabilities current LLMs are unable to meet due to their generic training and static reasoning mechanisms Toro et al. (2024); Liu et al. (2025b); Fathallah et al. (2024b). Moreover, while early-stage tasks in OE benefit from relatively clear and quantifiable optimization metrics, maintenance and evaluation activities often involve complex and less defined objectives, such as ensuring semantic coverage, consistency robustness, and ontology evolution over time.

Future research directions may include developing hybrid learning pipelines that allow LLMs to continuously integrate domain updates, such as new regulations or emerging scientific knowledge, thereby supporting ongoing validation and refinement based on domain-specific feedback. Incorporating neuro-symbolic architectures into LLMs and introducing additional context, such as explicit metrics for consistency robustness and evaluation reports and guidelines on ontology design, may improve their performance on OE tasks Idelfonso Magana Vsevolodovna and Monti (2025).

5.2 Configuration workflows of LLMs in ontology development activities

Our findings show that LLMs play multiple roles in ontology engineering, primarily as ontology engineers and domain experts, with an emerging potential to support ontology validation and refinement as evaluators. From the LLMs used, GPT-based models dominate in reasoning-intensive tasks, while open-source and lightweight models are increasingly favored for ontology matching and conceptualization, reflecting a diverse LLM ecosystem. By analyzing the inputs and outputs of LLM-supported OE tasks, we observe that LLMs can process a wide range of inputs, from unstructured natural language text to semi-structured and structured data, and generate outputs aligned with various stages of ontology engineering. These outputs include structured text, executable queries, and formal representations such as OWL axioms and validation mappings. Although natural language remains the predominant input modality, there is a clear shift toward adopting more structured input and output formats to better guide LLM behavior and produce machine-processable results, particularly in high-precision ontology engineering tasks. Despite these advances, human expertise remains essential. LLMs may still misunderstand domain-specific details.

Therefore, expert review, iterative improvements, and stakeholder participation are crucial throughout all stages of OE tasks to ensure the accuracy of the outputs.

Based on our analysis, the contributions of LLMs to OE processes can be summarized as follows: LLMs can effectively assume multiple roles within OE tasks, notably as ontology engineers and domain experts. In these roles, LLMs support the automation of ontology construction and the enrichment of domain-specific knowledge, substantially reducing the manual effort and transfer of domain-specific expertise traditionally required by ontology engineers.

Different model sizes have been used successfully for different tasks. For example, fine-tuned GPT-4 models have been shown to produce highly accurate and syntactically correct ontology fragments, while smaller models such as Mistral-7B, despite having fewer parameters, provide faster responses and efficient performance, particularly on smaller or domain-specific datasets Jindal et al. (2024); Ahuja et al. (2024).

The integration of LLMs into OE introduces fundamentally new paradigms and workflows compared to traditional practices. LLM-based approaches enable more conversational and iterative processes, exhibit emerging reasoning abilities and can directly allowing not only ontology engineers, but also non-experts to contribute by providing natural language text inputs to generate structured outputs. These inputs "feed" the LLMs, which in turn generate ontology snippets, suggest refinements, or offer validation feedback. This transformation fosters greater flexibility, accelerates iteration cycles, and enhances adaptability within ontology engineering workflows.

Despite their strengths, LLMs in OE tasks face limitations, notably resource requirements for access, fine-tuning, and infrastructure, which may restrict their accessibility and scalability Hoffmann et al. (2022); Kaddour et al. (2023); Treviso et al. (2023). Beyond that, current LLMs generalize poorly across specialized domains unless carefully guided by well-designed prompts. Therefore, effective prompt design becomes important to ensure that LLMs can correctly interpret domain-specific concepts and generate relevant outputs. Without such guidance, their outputs are often incomplete, ensure, or semantically irrelevant Barman et al. (2024); Ehsani et al. (2025). Also, compared to formal logic systems Baader et al. (2017); Heindorf et al. (2022), the reasoning abilities of LLMs remain relatively shallow Xu et al. (2025). They may generate hallucinated facts or relationships and offer limited transparency on how outputs are produced Huang et al. (2025). Moreover, LLMs often violate fundamental ontological constraints, such as class disjointness, hierarchical structures, and domain or range restrictions, due to the absence of internal consistency checks Petroni et al. (2019); West et al. (2021). Consequently, outputs generated by LLMs typically require external validation, post-processing, and expert correction to ensure logical soundness and semantic coherence within OE workflows.

To overcome present limitations of LLMs in OE, future research should pursue hybrid neuro-symbolic methods that blend the generative flexibility of language models with rule-based reasoners Servantez et al. (2024), thereby boosting logical soundness and semantic consistency West et al. (2021); Hitzler et al. (2022); Han et al. (2024). At the same time, more robust domain-adaptive prompt engineering techniques are needed to better steer models, reduce hallucination, and semantic drift Jayasuriya et al. (2025); Zhang et al. (2024b); Liu et al. (2025a). Building automated validation pipelines that combine formal consistency checks with expert-in-the-loop review would further enhance scalability and reliability.

Finally, enhancing the transparency of LLMs remains an open challenge Bommasani et al. (2021); Zhao et al. (2024). For example, enabling models to explain how each answer is generated and to trace the provenance of every produced result would not only increase user trust but also facilitate the future reuse and maintenance of outputs from OE activities.

Overall, achieving the full potential of LLMs in ontology engineering calls for technical advances in both models and workflows, along with stronger human oversight, richer domain expertise, and rigorous formal verification.

5.3 Evaluation gaps and challenges for LLMs in ontology development activities

Our review indicates that empirical validation is now a cornerstone of LLM-based OE research. Almost two-thirds of the surveyed papers include full experimental evaluations, usually built on open-source domain ontologies in OWL or RDF, which act as expert-curated benchmarks. Most studies adopt quantitative, qualitative, or hybrid methods, reporting metrics such as precision, recall, F1, or semantic similarity scores, and many complement these figures with expert judgments to assess conceptual soundness and domain relevance. When evaluating the performance reported in these studies, it is important to note that the assessments target entire pipelines rather than individual LLM-based components. Since each study adopts its own baselines and benchmarks, direct cross-paper performance comparisons are rarely fair or meaningful. Nevertheless, the evaluation processes across the literature consistently suggest that integrating LLMs contributes to enhanced automation and shows promising improvements in the outputs across multiple stages of the OE lifecycle.

Across the reviewed studies, there is a clear trend toward stronger adoption of open-source publicly available datasets, supporting the creation of more reproducible evaluation frameworks. Shared ontologies now may serve as common baselines that future research can replicate and extend. Several studies have deliberately adopted gold-standard datasets to ensure greater fairness and comparability across different approaches.

At the same time, early efforts toward standardized and transparent evaluation protocols have begun to emerge. Initiatives such as the OAEI [11] and the LLMs4OL Giglou et al. (2024) challenge explicitly define datasets, subtasks, and evaluation metrics, marking a move toward greater consistency and reproducibility within the field.

From an evaluation perspective, quantitative methods provide objective, reproducible, and scalable measurements for LLM-based ontology engineering, using metrics such as precision, recall, F1-score, and semantic similarity. Qualitative evaluations by domain experts capture semantic coherence, contextual relevance, and conceptual soundness that numerical metrics often miss. Combining both approaches leverages statistical rigor with semantic depth, enabling a more comprehensive and trustworthy assessment of LLM outputs. This integrated strategy ensures that generated ontologies are not only formally correct but also contextually meaningful and practically applicable

However, several limitations remain. We find that evaluation practices in LLM-based OE still lack standardized protocols. Most studies define their own tasks, datasets, metrics, and benchmarks, making it difficult to perform meaningful comparisons across different works. The experimental landscape remains highly heterogeneous, with differences in models, data, and evaluation criteria that make results rarely directly comparable, and even small variations such as prompt design or corpus selection can significantly influence results.

Another critical limitation is the conflation of LLM performance with the overall behavior of the pipeline. Many studies evaluate the final system output without clearly isolating the contributions of

[11] https://oaei.ontologymatching.org/

individual components, making it difficult to accurately assess the true capabilities and weaknesses of LLMs. Although combining quantitative metrics with expert evaluations has improved current practices, challenges remain. Quantitative metrics often fail to capture deeper semantic relationships and domain-specific subtleties, while qualitative assessments are time-consuming Queirós et al. (2017), require substantial domain expertise, and introduce subjectivity, thereby limiting scalability and generalization.

To address these limitations, future research should prioritize the development of standardized evaluation protocols for LLM-base OE. This includes creating unified benchmarks with clearly defined datasets, tasks, and metrics to facilitate more consistent comparisons across studies. Modular evaluation frameworks are also needed Wu and Yu (2024) to disentangle the specific contributions of LLMs from other pipeline components, enabling a clearer understanding of model-specific strengths and weaknesses. Furthermore, there is a need to refine the evaluation metrics, moving beyond surface-level accuracy scores toward measures that better capture semantic coherence, conceptual soundness, and domain relevance. Finally, new methods should be explored to enhance the depth and scalability of evaluations, such as employing semi-automated semantic validation tools and establishing standardized expert review processes.

5.4 Application domains of LLM-based ontology development

Across the reviewed studies, there is a clear trend toward a wider application of LLMs in OE. While early research focused on healthcare and life sciences, recent work shows growing adoption in fields such as cultural heritage, finance, emergency management, autonomous systems, and academic knowledge organization. This expansion underscores the scalability and adaptability of LLMs in supporting ontology tasks across diverse domains.

Life sciences Fathallah et al. (2024b) and healthcare Yang et al. (2023) remain methodologically mature, benefiting from rich gold-standard resources and established terminologies. In contrast, fields without consolidated vocabularies such as disaster response and finance are beginning to apply LLMs but often lack standardized workflows and benchmarks. LLMs offer notable advantages by lowering the barrier to ontology development, enabling domain experts to participate more directly in OE tasks, and supporting flexible, iterative construction from diverse textual inputs.

However, domain-specific adaptation remains challenging Mai et al. (2024), as LLMs trained on general corpora may miss specialized terminologies and evolving knowledge structures without careful prompt design. Ensuring semantic precision and formal consistency still requires substantial expert validation, especially in regulated domains Perera and Liu (2024). Furthermore, scalability problems arise because LLMs, being statically trained, cannot dynamically incorporate new knowledge without retraining, limiting their long-term applicability Du et al. (2023).

To address these challenges, future research should focus on developing domain-specific benchmarks, standardized evaluation frameworks, and hybrid workflows that combine LLM outputs with formal reasoning and expert validation. Further exploration into continual learning strategies Wu et al. (2024) and dynamic update mechanisms Fan et al. (2023) may enhance the sustainability and robustness of LLM-driven ontology engineering across domains.

6 Conclusion

Our systematic literature review of 41 experiments across 30 publications reveals both the promise and limitations of integrating LLMs into OE workflows. LLMs show particular strengths in early-stage

activities, such as domain conceptualization, requirements specification, and implementation, where they bridge natural language understanding with formal ontological structures. Models like GPT, LLaMA have been applied to generate CQs, formal axioms, documentation, and validation scripts, fulfilling roles as ontology engineers, domain experts, and evaluators. Their adaptability across domains such as healthcare, cultural heritage, and autonomous systems further illustrates their versatility.

Nevertheless, LLMs in OE tasks still face challenges. Their reasoning remains shallow, often leading to hallucinated facts and limited transparency. Support across the ontology lifecycle is uneven, with maintenance particularly underexplored. Evaluation practices are fragmented, as existing quantitative metrics fail to fully capture performance (especially in hybrid assessments), tasks present different inputs and outputs, and there is a lack of common benchmarks to enable the comparison of different approaches. LLMs also struggle with domain adaptation, scalability, and sensitivity to prompt design, limiting their real-world applicability in ontology evolution. In order to address these issues, we propose the following research challenges and suggestions:

- **Hybrid Neuro-Symbolic Reasoning**: Develop systems that combine LLM-generated suggestions with logic validation to enhance logical consistency and reduce hallucinations.
- **Lifecycle Coverage Expansion**: Extend LLM applications to underrepresented ontology lifecycle stages, particularly documentation, maintenance, and versioning, to support long-term ontology sustainability.
- **Standardized Evaluation Frameworks**: Establish reproducible benchmarks with open datasets and evaluation metrics that integrate both quantitative measures and domain-specific semantic assessments, enabling meaningful performance comparisons between different efforts.
- **Continual Learning and Dynamic Adaptation**: Design domain-adaptive LLMs capable of incorporating evolving knowledge without requiring complete retraining, improving scalability and relevance in dynamic domains.
- **Enhancing Real-World Robustness**: Refine prompt engineering methodologies and reduce reliance on structured inputs to strengthen LLM adaptability for practical OE scenarios.

By addressing these challenges, LLMs may evolve from task-specific tools to robust partners in collaborative ontology engineering, fostering scalable and high-quality knowledge representation across domains.

7 Appendix

This appendix presents additional material supporting the main text, including extended tables and detailed data referenced throughout the study.

7.1 LLM-supported Ontology Development Activities

Table 2 provides a comprehensive mapping of how LLMs contribute to specific ontology engineering activities across the 41 reviewed studies. Each row represents one distinct studies and several studies might belong to the same paper, that is, for the cases in which one paper uses LLMs to support more

than one activity. The table highlights the role, model used, input and output formats, and whether human participants were involved in the LLMs supported component.

Table 2. Details of LLM-supported ontology engineering activities, including the assigned roles of LLMs, model types used, input formats, generated outputs, and whether human involvement was required (indicated as YES/NO).

Resource	Role	Model	Inputs	Outputs	Human involved
Requirements Specification – Functional Requirements Writing					
Fathallah et al. (2024a)	Ontology Engineer	GPT-3.5, LLaMA, PaLM	Natural language text	Natural language text	NO
Antia and Keet (2023)	Ontology Engineer	T5	Natural language text	CQs	NO
Requirements Specification – CQ Reverse Engineering					
Rebboud et al. (2024a)	Domain Experts	LangChain, Ollama	Ontologies	CQs	NO
Alharbi et al. (2024a)	Ontology Engineer	GPT-3.5-turbo, GPT-4, LLaMA2	Triples	CQs	YES
Ciroku et al. (2024a)	Ontology Engineer	MiniLM, T5, SBERT	KGs	CQs	NO
Rebboud et al. (2024b)	Ontology Engineer	GPT-3.5, GPT-4	Ontologies	CQs	NO
Alharbi et al. (2024b)	Ontology Engineer	GPT-3.5-turbo, GPT-4, LLaMA-2-70B, Mistral 7B, Flan-T5-XL	Triples	CQs	NO
Requirements Specification – Requirement Formalization					
Rebboud et al. (2024a)	Ontology Engineer	LangChain, Ollama	Ontologies and CQs	Queries	NO
Tufek et al. (2024)	Ontology Engineer	ChatGPT	Natural language text or CQs	SPARQL Queries	NO
Kholmska et al. (2024)	Ontology Engineer	ChatGPT, Bard	Concepts	SPARQL Queries	NO
Ontology Implementation – Conceptualization					
Rebboud et al. (2024a)	Domain Experts	LangChain, Ollama	CQs	Ontologies	NO
Bischof et al. (2024)	Domain Experts	Mistral 7B	Natural language text	Terms	NO
Goyal et al. (2024)	Ontology Engineer	LLaMA3, GPT-4o, Mistral	Natural language text	Binary decision	NO
Coutinho (2024)	Ontology Engineer	Not mentioned	Natural language text	Summarization	NO
Kholmska et al. (2024)	Step 2:Ontology Engineer Step 3:Ontology Engineer	Step 2: ChatGPT, Bard Step 3: ChatGPT, Bard	Step 2:Natural language text Step 3:Natural language text	Step 2:Classes Step 3:Concepts	Step 2: NO Step 3: NO
Dong et al. (2024)	Domain Expert, Ontology Engineer	GPT-3.5, LLaMA2, FLAN-T5, GPT-4	Natural language text, Ontologies	Natural language text	NO
Babaei Giglou et al. (2023)	Ontology Engineer	BERT, BLOOM, LLaMA, GPT-3, GPT-3.5, GPT-4, BART, Flan-T5	Task A: Natural language text, lexical term Task B: Natural language text Task C: Natural language text	Task A: Term type Task B: Binary decision Task C: Binary decision	NO
Toro et al. (2024)	Ontology Engineer	GPT-4, GPT-3.5-turbo	Term	JSON or YAML	NO
Pisu et al. (2024)	Ontology Engineer	BERT	Nature language text	Relationships	NO
Ontology Implementation – Encoding					
Doumanas et al. (2024)	Ontology Engineer	GPT-4, GPT-3.5, Bard, LLaMA	Phase 1: Natural language text Phase 2: Domain documents Phase 3: Natural language text and CQs	Phase 1: Ontologies Phase 2: Ontologies Phase 3: Ontologies	YES
Fathallah et al. (2024a)	Ontology Engineer	GPT-3.5, LLaMA, PaLM	Natural language text	CQs, Triples and Ontologies	NO
Caufield et al. (2024)	Ontology Engineer	OntoGPT	Natural language text	Ontologies	NO
Eells et al. (2024)	Ontology Engineer	GPT-4	Natural language text	Natural language text	NO

Table 2

Resource	Role	Model	Inputs	Outputs	Human involved
Saeedizade and Blomqvist (2024)	Ontology Engineer	GPT-3.5, GPT-4, Bard, LLaMA-7B, LLaMA-13B, LLaMA2-70B, Alpaca, Falcon-7B, Falcon-7B-Instruct, WizardLM, Alpaca-LoRA	CQs	Ontologies	NO
Mateiu and Groza (2023)	Ontology Engineer	GPT-3, Davinci model	Natural language text	Axioms	NO
Tang et al. (2023)	Ontology Engineer	ChatGPT	Natural language text	Ontologies, JSON and Triples	NO
da Silva et al. (2024)	Ontology Engineer	GPT-4, Turbo4, Claude3, Gemini Pro	Natural language text, Ontologies	Ontologies	NO
Ontology Development – Ontology Matching and Reuse					
Zamazal (2024)	Domain Experts	GPT-4o	Natural language text and verbalized candidates	Binary decision	NO
Kholmska et al. (2024)	Ontology Engineer	GPT, Bard	Step 4: Natural language text Step 6: Concepts, Ontologies, Natural language text	Step 4: Documentation Step 6: Mapping	NO
Hertling and Paulheim (2023)	Ontology Engineer	LLaMA	Ontologies and Natural language text	Mapping	NO
He et al. (2023)	Ontology Engineer	Flan-T5-XXL, GPT-3.5-turbo	Natural language text	Binary decision	NO
Norouzi et al. (2023)	Ontology Engineer	ChatGPT	Natural language text	Mapping	NO
Ontology Development – Ontology Evaluation					
Tsaneva et al. (2024)	Domain Experts, Human Evaluator	GPT-4	Natural language text	Axioms	NO
Kholmska et al. (2024)	Ontology Engineer	GPT, Bard	Step 5: Ontologies	Step 5: Natural language text	NO
Fathallah et al. (2024a)	Domain Expert, Ontology Engineer	GPT-3.5, LLaMA, PaLM	Natural language text	Ontologies and Axioms	NO
Zhang et al. (2025)	Ontology Engineer	GPT-3	Ontologies and CQs	Binary decision	YES
Ontology Publication – Documentation					
Rebboud et al. (2024a)	Domain Experts	LangChain, Ollama	Ontologies	Documentation	NO
Kholmska et al. (2024)	Ontology Engineer	GPT, Bard	Step 9: Ontology Extensions, Natural language text	Step 9: Documentation	NO
Fathallah et al. (2024a)	Ontology Engineer	GPT-3.5, LLaMA, PaLM	Natural language text, Ontologies	Documentation	NO
Giri et al. (2024)	Domain Experts, Ontology Engineer	T5	Terms	Documentation	NO
Maintenance – Bug Issue					
Kholmska et al. (2024)	Domain Expert	GPT, Bard	Step 8: Natural language text	Step 8: Natural language text	YES

7.2 Experimental Setup and Evaluation Overview

Table 3 summarizes the experimental validation practices across all 41 reviewed studies. It records whether an experiment was performed, provides open-source access links when available, identifies the datasets utilized, and details the evaluation methodology (quantitative, qualitative, or mixed by both) along with the specific metrics employed. By including dataset sources and tool repositories, the table aims to support reproducibility and offers insights into the evaluation rigor and maturity within the field.

Table 3. Summary of experiments, data sources, evaluation types, and evaluation metrics used in LLM-supported ontology engineering studies.

Paper resource	Experiment	Data source	Evaluation type	Evaluation metric
Requirements specification – Functional requirements writing				

Table 3

Paper resource	Experiment	Data source	Evaluation type	Evaluation metric
Fathallah et al. (2024a)	YES , a test[34]	Wine	N/A	N/A
Alharbi et al. (2024a)	YES[12]	VideoGame, VICINITY Core, Dem@care	Hybrid	Numbers (CQs, triples), Precision, Recall, F1
Requirements specification – CQ Reverse Engineering				
Rebboud et al. (2024a)	YES[33]	DOREMUS, Polifonia, DemCare, Odeuropa, NORIA-O, FIBO	Quantitative	Cosine Similarity
Ciroku et al. (2024a)	YES[13]	WDV[14]	Quantitative	BLEU score
Rebboud et al. (2024b)	YES[33]	DOREMUS, Polifonia, Dem@Care, Odeuropa, NORIA-O, FIBO	Quantitative	Cosine Similarity
Antia and Keet (2023)	YES[15]	Covid19 articles[16]	Qualitative	Human comment
Alharbi et al. (2024b)	YES	VideoGame, Dem@care, VICINITY Core, African Wildlife	Quantitative	Precision, Recall, F1
Requirements specification – Requirement formalization				
Rebboud et al. (2024a)	YES[33]	DOREMUS, Polifonia, DemCare, Odeuropa, NORIA-O, FIBO	Quantitative	Tree Edit Distance
Tufek et al. (2024)	YES[17]	Smart Applications REFerence, OPC UA Robotics	Quantitative	Precision, Recall, F1
Kholmska et al. (2024)	N/A	OntoDM	Quantitative	Model Consistency Error Rate Reduction Coverage of Relevant Concepts
Ontology implementation – Conceptualization				
Rebboud et al. (2024a)	YES[33]	DOREMUS, Polifonia, DemCare, Odeuropa, NORIA-O, FIBO	Quantitative	Precision, Recall, F1, Accuracy, Consistent Ontology
Bischof et al. (2024)	N/A	N/A	Qualitative	Expert reviews
Goyal et al. (2024)	YES[18]	Task B: GeoNames, Schema.org, UMLS, GO Task C: UMLS	Quantitative	Precision, F1-score
Coutinho (2024)	N/A	UFO	Hybrid	Time, Model Quality Metrics, User Satisfaction, Domain Experts Feedback
Kholmska et al. (2024)	N/A	OntoDM	Quantitative	Inter-Model Consistency, Error Rate Reduction, Coverage of Relevant Concepts
Dong et al. (2024)	YES[19]	MM-S14-Disease, MM-S14-CPP	Quantitative	InRank@k, InRecall@k
Babaei Giglou et al. (2023)	YES[20]	WordNet, GeoNames, UMLS, National Cancer Institute, MEDCIN, SNOMEDCT US, Schema.org	Quantitative	MAP@K, F1-score
Toro et al. (2024)	YES[21]	Cell Ontology, UBERON, GO, Human Phenotype Ontology, Mammalian Phenotype Ontology, MONDO, Environment Ontology, Food Ontology, Ontology of Biomedical Investigations, Ontology of Biological Attributes	Quantitative and Qualitative	Accuracy, Recall, F1, Manual Assessment
Pisu et al. (2024)	YES[22]	Computer Science Ontology	Quantitative	Accuracy, Precision, Recall, F1
Ontology implementation – Encoding				
Doumanas et al. (2024)	YES[23]	Wildfire	Hybrid	Analysis of False Positives Precision, Recall, F1-score
Fathallah et al. (2024a)	YES, a test[34]	Wine	N/A	N/A
Caufield et al. (2024)	YES[24]	GO, EMAPA, MONDO Disease Ontology	Quantitative	F1, Precision, Recall
Eells et al. (2024)	YES[25]	101 nouns from COCA	N/A	N/A
Saeedizade and Blomqvist (2024)	YES[26]	Music, Theater, Hospital	Qualitative	Score Evaluation
Mateiu and Groza (2023)	N/A	150 sentences	N/A	N/A
Tang et al. (2023)	N/A	OpenXOntology	N/A	N/A

Table 3

Paper resource	Experiment	Data source	Evaluation type	Evaluation metric
da Silva et al. (2024)	YES[27]	CaSk	Quantitative and Qualitative	Mean Error Score
Ontology development – Ontology matching and reuse				
Zamazal (2024)	YES[28]	EDOAL, Manchester from OAIE	Hybrid	Precision, Relaxed Precision, Recall
Kholmska et al. (2024)	N/A	OntoDM	N/A	N/A
Hertling and Paulheim (2023)	YES[29]	Ontologies from OAIE	Quantitative	Precision, Recall, F1, Size, Time
He et al. (2023)	YES[30]	NCIT-DOID, SNOMED-FMA	Quantitative	Precision, Recall, F1, Hits, MRR, RR
Norouzi et al. (2023)	YES	Ontologies from OAIE	Quantitative	Precision, Recall, F1
Ontology development – Ontology evaluation				
Tsaneva et al. (2024)	YES	Pizza Ontology	Hybrid	Accuracy, Precision, Recall, F1, Majority Vote Aggregation
Kholmska et al. (2024)	N/A	OntoDM	N/A	N/A
Fathallah et al. (2024a)	YES, a test[34]	Wine	N/A	N/A
Zhang et al. (2025)	YES[31]	Music Meta	Qualitative	Feedback Scores
Ontology publication – Documentation				
Rebboud et al. (2024a)	YES[33]	DOREMUS, Polifonia, DemCare, Odeuropa, NORIA-O, FIBO	Quantitative	Cosine Similarity
Kholmska et al. (2024)	N/A	OntoDM	Quantitative	Inter-Model Consistency, Error Rate Reduction, Coverage of Relevant Concepts
Fathallah et al. (2024a)	YES, a test[34]	Wine	N/A	N/A
Giri et al. (2024)	YES[32]	GO	Quantitative and Qualitative	Correlation with Embedding Scores, Confidence Scores
Maintenance – Bug issue				
Kholmska et al. (2024)	N/A	OntoDM	Quantitative	Inter-Model Consistency, Error Rate Reduction, Coverage of Relevant Concepts

7.3 Application Domains of LLMs in OE

Table 4 presents a categorization of application domains where LLMs are used in ontology development. For each domain, we list representative ontologies and the key studies that utilized them across our review. This offers insights into how LLM applications vary across domains such as healthcare, cultural heritage, and autonomous systems.

Table 4. Key application domains, example ontologies, and representative studies using LLMs for ontology development.

Application Domain	Example Ontologies	Key Papers
Healthcare & Medicine	DemCare, SNOMED CT, UMLS	Tsaneva et al. (2024), He et al. (2023)
Cultural Heritage	DOREMUS, Polifonia, Odeuropa	Rebboud et al. (2024a,b), Zhang et al. (2025)
Finance & Banking	FIBO	Rebboud et al. (2024a,b)
Search & Rescue (SAR)	SAR Ontology	Doumanas et al. (2024)
Biology & Life Sciences	Gene Ontology (GO), MONDO	Giri et al. (2024), Caufield et al. (2024)
Autonomous Driving	Road traffic ontologies	Tang et al. (2023)
Education & Research	Computer Science Ontology	Pisu et al. (2024)
Food & Agriculture	FoodOn	Caufield et al. (2024)

7.4 Ontology Datasets Used Across Studies

Table 5 lists all experiment datasets utilized in the reviewed studies, with their corresponding names, access URLs, and associated domains. This compilation supports transparency and facilitates future replication or comparative benchmarking using the same datasets.

Table 5. Summary of experiment datasets used across the reviewed studies, including their names, access links, and corresponding application domains.

Acronym Name	Full Name	URL	Domain
African Wildlife	African Wildlife Ontology	http://www.meteck.org/teaching/ontologies-/AfricanWildlifeOntology1.owl	Ecology
CaSk	Capability and Skill Ontology	https://github.com/CaSkade-Automation/CaSkMan	Robotics
CL	Cell Ontology	https://github.com/obophenotype/cell-ontology	Anatomy
CSO	Computer Science Ontology	https://cso.kmi.open.ac.uk/home	Computer Science

[12] https://github.com/SemTech23/RETROFIT-CQs
[13] https://github.com/King-s-Knowledge-Graph-Lab/revont
[14] https://github.com/gabrielmaia7/WDV
[15] https://github.com/pymj/AgOCQs
[16] https://github.com/pymj/AgOCQs/tree/main/AgOCQs/inputText
[17] https://github.com/Siemens-OKE/llm-query-pipeline
[18] https://drive.google.com/drive/folders/1vRynlNH6LouIvcI1ymHsm6DwYKSOUoAa
[19] https://github.com/KRR-Oxford/LM-ontology-concept-placement
[20] https://github.com/HamedBabaei/LLMs4OL
[21] https://github.com/monarch-initiative/dragon-ai-results
[22] https://github.com/aleessiap/LeveragingLMforGeneratingOntologies
[23] https://github.com/dimitrisdoumanas19/New-Experiments-LLMs.git
[24] https://github.com/monarch-initiative/ontogpt
[25] https://github.com/kastle-lab/commonsense-micropatterns
[26] https://github.com/LiUSemWeb/LLMs4OntologyDev-ESWC2024
[27] https://github.com/CaSkade-Automation/llm-capability-generation
[28] https://github.com/OndrejZamazal/ComplexOntologyMatching-SEMANTiCS2024
[29] https://figshare.com/articles/code/OLaLa_for_OAEI
[30] https://github.com/KRR-Oxford/LLMap-Prelim
[31] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat
[32] https://github.com/kiharalab/GO2Sum

Table 5

Acronym Name	Full Name	URL	Domain
DemCare	Dementia Care Ontology	https://demcare.eu/ontologies	Healthcare
DOREMUS	Music Ontology	http://data.doremus.org/ontology	Arts
EMAPA	Mouse Developmental Anatomy	https://obofoundry.org/ontology/emapa.html	Anatomy
ENVO	Environment Ontology	https://github.com/EnvironmentOntology/envo	Environment
FIBO	Financial Industry Business Ontology	https://github.com/edmcouncil/fibo	Business
FOODON	Food Ontology	https://github.com/FoodOntology/foodon	Food
GO	Gene Ontology	http://geneontology.org	Biology
HP	Human Phenotype Ontology	https://github.com/obophenotype/human-phenotype-ontology	Phenotype
MONDO	Mondo Disease Ontology	https://github.com/monarch-initiative/mondo	Disease
MP	Mammalian Phenotype Ontology	https://github.com/obophenotype/mammalian-phenotype-ontology	Phenotype
MusicMeta	Music Metadata Ontology	https://w3id.org/polifonia/ontology/music-meta	Music
NORIA-O	Norwegian AI Ontology	https://w3id.org/noria	AI
OAIE	Ontology Alignment Evaluation Initiative	https://oaei.ontologymatching.org	Benchmark
OBA	Ontology of Biological Attributes	https://github.com/obophenotype/biological-attributes-ontology	Attributes
OBI	Ontology for Biomedical Investigations	https://github.com/obi-ontology/obi	Methodology
Odeuropa	Olfactory Heritage Ontology	https://odeuropa.eu	Cultural Heritage
OntoDM	Ontology of Data Mining	https://lod-cloud.net/dataset/bioportal-ontodm	Data Science
OpenXOntology	Open Exchange Ontology	https://openxontology.org	Business
OPC-UA	OPC Unified Architecture	https://github.com/OPCFoundation/UA-Nodeset	Industrial
Polifonia	Polifonia Ontology Network	https://github.com/polifonia-project	Music
SAREF	Smart Appliances Reference Ontology	https://saref.etsi.org	IoT
UBERON	Uberon Multi-species Anatomy Ontology	https://github.com/obophenotype/uberon	Anatomy
UFO	Unified Foundational Ontology	https://ontouml.readthedocs.io/en/latest/intro/ufo.html	Foundational
UMLS	Unified Medical Language System	https://www.nlm.nih.gov/research/umls	Medicine
VICINITY	IoT Core Ontology	http://iot.linkeddata.es/def/core	IoT
WDV	Web Data Vocabulary	https://github.com/gabrielmaia7/WDV	Web
Pizza	Pizza Ontology	https://protege.stanford.edu/ontologies/pizza-/pizza.owl	Food
NCIT	National Cancer Institute Thesaurus	https://bioportal.bioontology.org/ontologies/NCIT	Oncology
DOID	Human Disease Ontology	https://bioportal.bioontology.org/ontologies/DOID	Disease
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms	https://www.snomed.org/	Medicine
FMA	Foundational Model of Anatomy	https://bioportal.bioontology.org/ontologies/FMA	Anatomy
MEDCIN	MEDCIN Ontology	https://www.sciencedirect.com/topics/nursing-and-health-professions/medical-ontology	Medicine
SNOMEDCT US	SNOMED CT United States Edition	https://www.nlm.nih.gov/healthit/snomedct/	Medicine
Schema.org	Schema.org Vocabulary	https://schema.org/	Web
Video Game Ontology	Video Game Ontology	https://vocab.linkeddata.es/vgo/	Entertainment
MM-S14-Disease/CPP	MM-S14-Disease/CPP Dataset	https://zenodo.org/records/10432003	Medicine
Wine Ontology	Wine Ontology	https://github.com/UCDavisLibrary/wine-ontology/blob/master/wine-ontology.owl	Food

References

Achichi M, Lisena P, Todorov K, Troncy R and Delahousse J (2018) Doremus: A graph of linked musical works. In: *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part II 17*. Springer, pp. 3–19.

Ahuja S, Tanmay K, Chauhan HH, Patra B, Aggarwal K, Del Corro L, Mitra A, Dhamecha TI, Awadallah A, Choudhary M et al. (2024) sphinx: Sample efficient multilingual instruction fine-tuning through n-shot guided prompting. *arXiv preprint arXiv:2407.09879* .

Alharbi R, Tamma V, Grasso F and Payne T (2024a) An experiment in retrofitting competency questions for existing ontologies. In: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. pp. 1650–1658. DOI:10.1145/3605098.3636053.

Alharbi R, Tamma V, Grasso F and Payne TR (2024b) Investigating open source llms to retrofit competency questions in ontology engineering. In: *Proceedings of the AAAI Symposium Series*, volume 4. pp. 188–198.

Amaral G, Rodrigues O and Simperl E (2022) WDV: A broad data verbalisation dataset built from wikidata. In: *International Semantic Web Conference*. Springer, pp. 556–574.

Antia MJ and Keet CM (2023) Automating the generation of competency questions for ontologies with agocqs. In: *Iberoamerican Knowledge Graphs and Semantic Web Conference*. Springer, pp. 213–227.

Baader F, Horrocks I, Lutz C and Sattler U (2017) *An introduction to description logic*. Cambridge University Press. DOI:https://doi.org/10.1017/9781139025355.

Babaei Giglou H, D'Souza J and Auer S (2023) Llms4ol: Large language models for ontology learning. In: *International Semantic Web Conference*. Springer, pp. 408–427.

Barman KG, Wood N and Pawlowski P (2024) Beyond transparency and explainability: on the need for adequate and contextualized user guidelines for LLM use. *Ethics and Information Technology* 26(3): 47.

Bennett M (2013) The financial industry business ontology: Best practice for big data. *Journal of Banking Regulation* 14(3): 255–268.

Bischof S, Filtz E, Parreira JX and Steyskal S (2024) Llm-based guided generation of ontology term definitions. In: *European Semantic Web Conference*. Springer, pp. 133–137.

Bittner T, Donnelly M and Winter S (2005) Ontology and semantic interoperability. In: *Large-scale 3D data integration*. CRC Press, pp. 139–160.

Bodenreider O (2004) The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research* 32(suppl_1): D267–D270.

Boell SK and Cecez-Kecmanovic D (2014) A hermeneutic approach for conducting literature reviews and literature searches. *Communications of the Association for information Systems* 34(1): 12.

Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, Bernstein MS, Bohg J, Bosselut A, Brunskill E et al. (2021) On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* .

Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I and Amodei D (2020a) Language models are few-shot learners. URL https://arxiv.org/abs/2005.14165.

[33] https://github.com/D2KLab/llm4ke
[34] https://github.com/andreamust/NEON-GPT

Brown TB et al. (2020b) Language models are few-shot learners. *arXiv:2005.14165* .

Carrera-Rivera A, Ochoa W, Larrinaga F and Lasa G (2022) How-to conduct a systematic literature review: A quick guide for computer science research. *MethodsX* 9: 101895.

Caufield JH, Hegde H, Emonet V, Harris NL, Joachimiak MP, Matentzoglu N, Kim H, Moxon S, Reese JT, Haendel MA et al. (2024) Structured prompt interrogation and recursive extraction of semantics (spires): A method for populating knowledge bases using zero-shot learning. *Bioinformatics* 40(3): btae104.

Chaplot DS (2023) Albert q. jiang, alexandre sablayrolles, arthur mensch, chris bamford, devendra singh chaplot, diego de las casas, florian bressand, gianna lengyel, guillaume lample, lucile saulnier, lélio renard lavaud, marie-anne lachaux, pierre stock, teven le scao, thibaut lavril, thomas wang, timothée lacroix, william el sayed. *arXiv preprint arXiv:2310.06825* .

Ciatto G, Agiollo A, Magnini M and Omicini A (2025) Large language models as oracles for instantiating ontologies with domain-specific knowledge. *Knowledge-Based Systems* 310: 112940.

Ciroku F, de Berardinis J, Kim J, Meroño-Peñuela A, Presutti V and Simperl E (2024a) Revont: Reverse engineering of competency questions from knowledge graphs via language models. *Journal of Web Semantics* 82(1): 100822.

Ciroku F, de Berardinis J, Kim J, Meroño-Peñuela A, Presutti V and Simperl E (2024b) RevOnt: Reverse engineering of competency questions from knowledge graphs via language models. *Journal of Web Semantics* : 100822.

Consortium GO (2006) The gene ontology (go) project in 2006. *Nucleic acids research* 34(suppl_1): D322–D326.

Coutinho ML (2024) Leveraging llms in text-based ontology-driven conceptual modeling .

da Silva LMV, Kocher A, Gehlhoff F and Fay A (2024) On the use of large language models to generate capability ontologies. In: *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, pp. 1–8.

Davies M (2010) The corpus of contemporary american english as the first reliable monitor corpus of english. *Literary and linguistic computing* 25(4): 447–464.

de Berardinis J, Carriero VA, Jain N, Lazzari N, Meroño-Peñuela A, Poltronieri A and Presutti V (2023) The polifonia ontology network: Building a semantic backbone for musical heritage. In: *International Semantic Web Conference*. Springer, pp. 302–322.

De Vergara JEL, Villagrá VA and Berrocal J (2004) Applying the web ontology language to management information definitions. *IEEE Communications Magazine* 42(7): 68–74. DOI:10.1109/MCOM.2004.1316535.

DeepMind G (2023) Gemini: A family of highly capable multimodal models. *arXiv:2312.11805* .

Diehl AD, Meehan TF, Bradford YM, Brush MH, Dahdul WM, Dougall DS, He Y, Osumi-Sutherland D, Ruttenberg A, Sarntivijai S et al. (2016) The cell ontology 2016: enhanced content, modularization, and ontology interoperability. *Journal of biomedical semantics* 7: 1–10.

Dimitropoulos K and Hatzilygeroudis I (2024) An ontology-knowledge graph based context representation scheme for robotic problems. In: *Proceedings of the 13th Hellenic Conference on Artificial Intelligence*. pp. 1–7.

Dong H, Chen J, He Y, Gao Y and Horrocks I (2024) A language model based framework for new concept placement in ontologies. In: Meroño Peñuela A, Dimou A, Troncy R, Hartig O, Acosta M, Alam M, Paulheim H and Lisena P (eds.) *The Semantic Web*. Cham: Springer Nature Switzerland. ISBN 978-3-031-60626-7, pp. 79–99.

Dong H, Chen J, He Y and Horrocks I (2023) Ontology enrichment from texts: A biomedical dataset for concept discovery and placement. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. pp. 5316–5320.

Donnelly K et al. (2006) Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics* 121: 279.

Doumanas D, Soularidis A, Kotis K and Vouros G (2024) Integrating llms in the engineering of a sar ontology. In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, pp. 360–374.

Drobnjakovic M, Kulvatunyou B, Ameri F, Will C, Smith B and Jones A (2022) The industrial ontologies foundry (iof) core ontology .

Du M, Luu AT, Ji B and Ng Sk (2023) From static to dynamic: A continual learning framework for large language models. *arXiv preprint arXiv:2310.14248* .

Eells A, Dave B, Hitzler P and Shimizu C (2024) Commonsense ontology micropatterns. In: *International Conference on Neural-Symbolic Learning and Reasoning*. Springer, pp. 51–59.

Ehsani R, Pathak S and Chatterjee P (2025) Towards detecting prompt knowledge gaps for improved LLM-guided issue resolution. *arXiv preprint arXiv:2501.11709* .

Fan L, Hua W, Li L, Ling H and Zhang Y (2023) Nphardeval: Dynamic benchmark on reasoning ability of large language models via complexity classes. *arXiv preprint arXiv:2312.14890* .

Fathallah N, Das A, Giorgis SD, Poltronieri A, Haase P and Kovriguina L (2024a) Neon-gpt: a large language model-powered pipeline for ontology learning. In: *European Semantic Web Conference*. Springer, pp. 36–50.

Fathallah N, Staab S and Algergawy A (2024b) Llms4life: Large language models for ontology learning in life sciences. *arXiv preprint arXiv:2412.02035* .

Fernández-Izquierdo A, Poveda-Villalón M and García-Castro R (2019) CORAL: a corpus of ontological requirements annotated with lexico-syntactic patterns. In: *The Semantic Web: 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2–6, 2019, Proceedings 16*. Springer, pp. 443–458.

Fernández-López M, Gómez-Pérez A and Juristo N (1997) Methontology: from ontological art towards ontological engineering .

Funk M, Hosemann S, Jung JC and Lutz C (2023) Towards ontology construction with language models. *arXiv preprint arXiv:2309.09898* .

Gangemi A and Presutti V (2009) Ontology design patterns. In: *Handbook on ontologies*. Springer, pp. 221–243.

Garijo D, Poveda-Villalón M, Amador-Dominguez E, Wang Z, García-Castro R and Corcho O (2024) Llms for ontology engineering: A landscape of tasks and benchmarking challenges. In: *Proceedings of the Special Session on Harmonising Generative AI and Semantic Web Technologies (HGAIS 2024) co-located with the 23rd International Semantic Web Conference (ISWC 2024)*. URL https://ceur-ws.org/Vol-3953/364.pdf.

Giglou HB, D'Souza J, Sadruddin S and Auer S (2024) Llms4ol 2024 datasets: Toward ontology learning with large language models. In: *Open Conference Proceedings*, volume 4. pp. 17–30.

Giri SJ, Ibtehaz N and Kihara D (2024) Go2sum: generating human-readable functional summary of proteins from go terms. *npj Systems Biology and Applications* 10(1): 29.

Glauer M, Memariani A, Neuhaus F, Mossakowski T and Hastings J (2024) Interpretable ontology extension in chemistry. *Semantic Web* 15(4): 937–958.

Gómez-Pérez A (1999) Ontological engineering: A state of the art. *Expert Update: Knowledge Based Systems and Applied Artificial Intelligence* 2(3): 33–43.

Goyal PK, Singh S and Tiwary US (2024) silp_nlp at llms4ol 2024 tasks a, b, and c: Ontology learning through prompts with llms. In: *Open Conference Proceedings*, volume 4. pp. 31–38.

Goyal T, Li JJ and Durrett G (2022) News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356* .

Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, Letman A, Mathur A, Schelten A, Vaughan A et al. (2024) The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* .

Guha RV, Brickley D and Macbeth S (2016) Schema.org: evolution of structured data on the web. *Commun. ACM* 59(2): 44–51. DOI:10.1145/2844544.

Guizzardi G, Wagner G, Almeida JPA and Guizzardi RS (2015) Towards ontological foundations for conceptual modeling: The unified foundational ontology (ufo) story. *Applied ontology* 10(3-4): 259–271.

Han S, Liu T, Li C, Xiong X and Cohan A (2024) Hybridmind: Meta selection of natural language and symbolic language for enhanced LLM reasoning. *arXiv e-prints* : arXiv–2409.

He Y, Chen J, Dong H and Horrocks I (2023) Exploring large language models for ontology alignment. *arXiv preprint arXiv:2309.07172* .

Heindorf S, Blübaum L, Düsterhus N, Werner T, Golani VN, Demir C and Ngonga Ngomo AC (2022) Evolearner: Learning description logics with evolutionary algorithms. In: *Proceedings of the ACM Web Conference 2022*. pp. 818–828.

Hertling S and Paulheim H (2023) Olala: Ontology matching with large language models. In: *Proceedings of the 12th Knowledge Capture Conference 2023*. pp. 131–139.

Hitzler P, Eberhart A, Ebrahimi M, Sarker MK and Zhou L (2022) Neuro-symbolic approaches in artificial intelligence. *National Science Review* 9(6): nwac035.

Hochreiter S and Schmidhuber J (1997) Long short-term memory. *Neural computation* 9(8): 1735–1780.

Hoffmann J, Borgeaud S, Mensch A, Buchatskaya E, Cai T, Rutherford E, Casas DdL, Hendricks LA, Welbl J, Clark A et al. (2022) Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* .

Hogan A, Blomqvist E, Cochez M, d'Amato C, Melo GD, Gutierrez C, Kirrane S, Gayo JEL, Navigli R, Neumaier S et al. (2021) Knowledge graphs. *ACM Computing Surveys (Csur)* 54(4): 1–37.

Huang J and Chang KCC (2022) Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403* .

Huang L, Yu W, Ma W, Zhong W, Feng Z, Wang H, Chen Q, Peng W, Feng X, Qin B et al. (2025) A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43(2): 1–55.

Hull D, Pettifer SR and Kell DB (2008) Defrosting the digital library: bibliographic tools for the next generation web. *PLoS computational biology* 4(10): e1000204.

Idelfonso Magana Vsevolodovna R and Monti M (2025) Enhancing large language models through neuro-symbolic integration and ontological reasoning. *arXiv e-prints* : arXiv–2504.

Jayasuriya D, Tayebati S, Ettori D, Krishnan R and Trivedi AR (2025) Sparc: Subspace-aware prompt adaptation for robust continual learning in LLMs. *arXiv preprint arXiv:2502.02909* .

Jiang X, Dong Y, Wang L, Fang Z, Shang Q, Li G, Jin Z and Jiao W (2024) Self-planning code generation with large language models. *ACM Transactions on Software Engineering and Methodology* 33(7): 1–30.

Jindal AK, Rajpoot PK and Parikh A (2024) Birbal: An efficient 7b instruct-model fine-tuned with curated datasets. *arXiv preprint arXiv:2403.02247* .

Joachimiak MP, Miller MA, Caufield JH, Ly R, Harris NL, Tritt A, Mungall CJ and Bouchard KE (2024) The artificial intelligence ontology: LLM-assisted construction of ai concept hierarchies. *Applied Ontology* : 15705838241304103.

Johnsen M (2025) *Developing AI Applications With Large Language Models*. Maria Johnsen.

Kaddour J, Harris J, Mozes M, Bradley H, Raileanu R and McHardy R (2023) Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169* .

Karakostas A, Briassouli A, Avgerinakis K, Kompatsiaris I and Tsolaki M (2016) The dem@ care experiments and datasets: a technical report. *arXiv preprint arXiv:1701.01142* .

Keet CM (2019) The african wildlife ontology tutorial ontologies. *Journal of Biomedical Semantics* 11.

Keet CM, Mahlaza Z and Antia MJ (2019) Claro: a data-driven cnl for specifying competency questions. *arXiv preprint arXiv:1907.07378* .

Kelly J, Sadeghieh T and Adeli K (2014) Peer review in scientific publications: benefits, critiques, & a survival guide. *Ejifcc* 25(3): 227.

Kheiri K and Karimi H (2023) Sentimentgpt: Exploiting gpt for advanced sentiment analysis and its departure from current machine learning. *arXiv preprint arXiv:2307.10234* .

Kholmska G, Kenda K and Rozanec J (2024) Enhancing ontology engineering with LLMs: From search to active learning extensions. *Proceedings of Data Mining and Data Warehauses – Sikdd 2024* .

Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J and Linkman S (2009) Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology* 51(1): 7–15.

Kotis KI, Vouros GA and Spiliotopoulos D (2020) Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations. *The Knowledge Engineering Review* 35: e4.

Krötzsch M and Thost V (2016) Ontologies for knowledge graphs: Breaking the rules. In: *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15*. Springer, pp. 376–392.

Lippolis AS, Ceriani M, Zuppiroli S and Nuzzolese AG (2024) Ontogenia: Ontology generation with metacognitive prompting in large language models. In: *European Semantic Web Conference*. Springer, pp. 259–265.

Lippolis AS, Saeedizade MJ, Keskisärkkä R, Zuppiroli S, Ceriani M, Gangemi A, Blomqvist E and Nuzzolese AG (2025) Ontology generation using large language models. *arXiv preprint arXiv:2503.05388* .

Lisena P, Schwabe D, van Erp M, Troncy R, Tullett W, Leemans I, Marx L and Ehrich SC (2022) Capturing the semantics of smell: the odeuropa data model for olfactory heritage information. In: *European Semantic Web Conference*. Springer, pp. 387–405.

Liu Y, Yang Q, Tang J, Guo T, Wang C, Li P, Xu S, Gao X, Li Z, Liu J et al. (2025a) Reducing hallucinations of large language models via hierarchical semantic piece. *Complex & Intelligent Systems* 11(5): 1–19.

Liu Z, Gan C, Wang J, Zhang Y, Bo Z, Sun M, Chen H and Zhang W (2025b) OntoTune: Ontology-driven self-training for aligning large language models. In: *Proceedings of the ACM on Web Conference 2025*. pp. 119–133.

Lo A, Jiang AQ, Li W and Jamnik M (2024) End-to-end ontology learning with large language models. *arXiv preprint arXiv:2410.23584* .

Mai HT, Chu CX and Paulheim H (2024) Do LLMs really adapt to domains? an ontology learning perspective. In: *International Semantic Web Conference*. Springer, pp. 126–143.

Masa P, Meditskos G, Kintzios S, Vrochidis S and Kompatsiaris I (2022) Ontology-based modelling and reasoning for forest fire emergencies in resilient societies. In: *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*. pp. 1–9.

Mateiu P and Groza A (2023) Ontology engineering with large language models. In: *2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE, pp. 226–229.

Mienye ID, Swart TG and Obaido G (2024) Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information* 15(9): 517.

Miller GA (1995) Wordnet: a lexical database for english. *Communications of the ACM* 38(11): 39–41.

Mukanova A, Milosz M, Dauletkaliyeva A, Nazyrova A, Yelibayeva G, Kuzin D and Kussepova L (2024) LLM-powered natural language text processing for ontology enrichment. *Applied Sciences* 14(13). DOI: 10.3390/app14135860. URL https://www.mdpi.com/2076-3417/14/13/5860.

Nakano R, Hilton J, Balaji S, Wu J, Ouyang L, Kim C, Hesse C, Jain S, Kosaraju V, Saunders W et al. (2021) Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* .

Norouzi SS, Mahdavinejad MS and Hitzler P (2023) Conversational ontology alignment with chatgpt. *arXiv preprint arXiv:2308.09217* .

Noy NF and McGuinness DL (2001) Ontology development 101: A guide to creating your first ontology. URL http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf.

OpenAI, Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, Almeida D, Altenschmidt J, Altman S, Anadkat S, Avila R, Babuschkin I, Balaji S, Balcom V, Baltescu P, Bao H, Bavarian M, Belgum J, Bello I, Berdine J, Bernadett-Shapiro G, Berner C, Bogdonoff L, Boiko O, Boyd M, Brakman AL, Brockman G, Brooks T, Brundage M, Button K, Cai T, Campbell R, Cann A, Carey B, Carlson C, Carmichael R, Chan B, Chang C, Chantzis F, Chen D, Chen S, Chen R, Chen J, Chen M, Chess B, Cho C, Chu C, Chung HW, Cummings D, Currier J, Dai Y, Decareaux C, Degry T, Deutsch N, Deville D, Dhar A, Dohan D, Dowling S, Dunning S, Ecoffet A, Eleti A, Eloundou T, Farhi D, Fedus L, Felix N, Fishman SP, Forte J, Fulford I, Gao L, Georges E, Gibson C, Goel V, Gogineni T, Goh G, Gontijo-Lopes R, Gordon J, Grafstein M, Gray S, Greene R, Gross J, Gu SS, Guo Y, Hallacy C, Han J, Harris J, He Y, Heaton M, Heidecke J, Hesse C, Hickey A, Hickey W, Hoeschele P, Houghton B, Hsu K, Hu S, Hu X, Huizinga J, Jain S, Jain S, Jang J, Jiang A, Jiang R, Jin H, Jin D, Jomoto S, Jonn B, Jun H, Kaftan T, Łukasz Kaiser, Kamali A, Kanitscheider I, Keskar NS, Khan T, Kilpatrick L, Kim JW, Kim C, Kim Y, Kirchner JH, Kiros J, Knight M, Kokotajlo D, Łukasz Kondraciuk, Kondrich A, Konstantinidis A, Kosic K, Krueger G, Kuo V, Lampe M, Lan I, Lee T, Leike J, Leung J, Levy D, Li CM, Lim R, Lin M, Lin S, Litwin M, Lopez T, Lowe R, Lue P, Makanju A, Malfacini K, Manning S, Markov T, Markovski Y, Martin B, Mayer K, Mayne A, McGrew B, McKinney SM, McLeavey C, McMillan P, McNeil J, Medina D, Mehta A, Menick J, Metz L, Mishchenko A, Mishkin P, Monaco V, Morikawa E, Mossing D, Mu T, Murati M, Murk O, Mély D, Nair A, Nakano R, Nayak R, Neelakantan A, Ngo R, Noh H, Ouyang L, O'Keefe C, Pachocki J, Paino A, Palermo J, Pantuliano A, Parascandolo G, Parish J, Parparita E, Passos A, Pavlov M, Peng A, Perelman A, de Avila Belbute Peres F, Petrov M, de Oliveira Pinto HP, Michael, Pokorny, Pokrass M, Pong VH, Powell T, Power A, Power B, Proehl E, Puri R, Radford A, Rae J, Ramesh A, Raymond C, Real F, Rimbach K, Ross C, Rotsted B, Roussez H, Ryder N, Saltarelli M, Sanders T, Santurkar S, Sastry G, Schmidt H, Schnurr D, Schulman J, Selsam D, Sheppard K, Sherbakov T, Shieh J, Shoker S, Shyam P, Sidor S, Sigler E, Simens M, Sitkin J, Slama K, Sohl I, Sokolowsky B, Song Y, Staudacher N, Such FP, Summers N, Sutskever I, Tang J, Tezak N, Thompson MB, Tillet P, Tootoonchian A, Tseng E, Tuggle P, Turley N, Tworek J, Uribe JFC, Vallone A, Vijayvergiya A, Voss C, Wainwright C, Wang JJ, Wang A, Wang B, Ward J, Wei J, Weinmann C, Welihinda A, Welinder P, Weng J, Weng L, Wiethoff M, Willner D, Winter C, Wolrich S, Wong H, Workman L, Wu S, Wu J, Wu M, Xiao K, Xu T, Yoo S, Yu K, Yuan Q, Zaremba W, Zellers R, Zhang C, Zhang M, Zhao S, Zheng T, Zhuang J, Zhuk W and Zoph B (2024) Gpt-4 technical report. URL https://arxiv.org/abs/2303.08774.

Panov P, Džeroski S and Soldatova L (2008) Ontodm: An ontology of data mining. In: *2008 IEEE International Conference on Data Mining Workshops*. IEEE, pp. 752–760.

Patel A and Debnath NC (2024) A comprehensive overview of ontology: Fundamental and research directions. *Current Materials Science: Formerly: Recent Patents on Materials Science* 17(1): 2–20. DOI:10.2174/2666145415666220914114301.

Perera O and Liu J (2024) Exploring large language models for ontology learning .

Petroni F, Rocktäschel T, Lewis P, Bakhtin A, Wu Y, Miller AH and Riedel S (2019) Language models as knowledge bases? *arXiv preprint arXiv:1909.01066* .

Pinto HS, Staab S and Tempich C (2004) Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In: *ECAI*, volume 16. p. 393.

Pisu A, Pompianu L, Salatino A, Osborne F, Riboni D, Motta E and Recupero DR (2024) Leveraging language models for generating ontologies of research topics. *Text2KG 2024: International Workshop on Knowledge Graph Generation from Text* URL https://ceur-ws.org/Vol-3747/text2kg%5fpaper6.pdf.

Poveda-Villalón M, Gómez-Pérez A and Suárez-Figueroa MC (2014) Oops! (ontology pitfall scanner!): An on-line tool for ontology evaluation. *Int. J. Semantic Web Inf. Syst.* 10(2): 7–34. DOI:10.4018/ijswis.2014040102. URL https://doi.org/10.4018/ijswis.2014040102.

Poveda-Villalón M, Fernández-Izquierdo A, Fernández-López M and García-Castro R (2022) Lot: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence* 111: 104755. DOI:https://doi.org/10.1016/j.engappai.2022.104755.

Queirós A, Faria D and Almeida F (2017) Strengths and limitations of qualitative and quantitative research methods. *European journal of education studies* .

Radford A, Narasimhan K, Salimans T and Sutskever I (2018) Improving language understanding by generative pre-training .

Radford A, Wu J, Child R, Luan D, Amodei D and Sutskever I (2019) Language models are unsupervised multitask learners. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Rebboud Y, Lisena P, Tailhardat L and Troncy R (2024a) Benchmarking llm-based ontology conceptualization: A proposal. In: *ISWC 2024, 23rd International Semantic Web Conference*.

Rebboud Y, Tailhardat L, Lisena P and Troncy R (2024b) Can llms generate competency questions? In: *ESWC 2024, Extended Semantic Web Conference*.

Rumelhart DE, Hinton GE and Williams RJ (1986) Learning representations by back-propagating errors. *nature* 323(6088): 533–536.

Saeedizade MJ and Blomqvist E (2024) Navigating ontology development with large language models. In: *European Semantic Web Conference*. Springer, pp. 143–161. DOI:10.1007/978-3-031-60626-7_8.

Sahbi A, Alec C and Beust P (2024) Automatic ontology population from textual advertisements: LLM vs. semantic approach. *Procedia Computer Science* 246: 3083–3092.

Salamon JS and Barcellos MP (2022) Towards a framework for continuous ontology engineering. In: *ONTOBRAS*. pp. 158–165.

Servantez S, Barrow J, Hammond K and Jain R (2024) Chain of logic: Rule-based reasoning with large language models. *arXiv preprint arXiv:2402.10400* .

Singh M, Cambronero J, Gulwani S, Le V and Verbruggen G (2023) Assessing gpt4-v on structured reasoning tasks. *arXiv preprint arXiv:2312.11524* .

Staab S, Studer R, Antoniou G and Van Harmelen F (2004) *OWL Web Ontology Language*. Ubiquity. Springer. ISBN 9781605660264, pp. 1–1. DOI:10.1145/1295289.1295290.

Staab S, Studer R, Schnurr HP and Sure Y (2001) Knowledge processes and ontologies. *IEEE Intelligent systems* 16(1): 26–34.

Studer R, Benjamins VR and Fensel D (1998) Knowledge engineering: Principles and methods. *Data & knowledge engineering* 25(1-2): 161–197.

Suárez-Figueroa MC, Gómez-Pérez A and Fernández-López M (2012) The neon methodology for ontology engineering : 9–34DOI:10.1007/978-3-642-24794-1_2.

Suárez-Figueroa MC, Gómez-Pérez A and Villazón-Terrazas B (2009) How to write and use the ontology requirements specification document. In: *On the Move to Meaningful Internet Systems: OTM 2009: Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009, Vilamoura, Portugal, November 1-6, 2009, Proceedings, Part II*. Springer, pp. 966–982.

Suárez-Figueroa MC and Gómez-Pérez A (2008) Towards a glossary of activities in the ontology engineering field. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco: European Language Resources Association (ELRA). ISBN 2-9517408-4-0. Http://www.lrec-conf.org/proceedings/lrec2008/.

Tailhardat L, Chabot Y and Troncy R (2024) Noria-o: an ontology for anomaly detection and incident management in ict systems. In: *European Semantic Web Conference*. Springer, pp. 21–39.

Tan H, Kebede R, Moscati A and Johansson P (2024) Semantic interoperability using ontologies and standards for building product properties. In: *12th Linked Data in Architecture and Construction Workshop, Bochum, Germany, June 13-14, 2024*. CEUR-WS, pp. 23–35.

Tang Y, Da Costa AAB, Zhang X, Patrick I, Khastgir S and Jennings P (2023) Domain knowledge distillation from large language model: An empirical study in the autonomous driving domain. In: *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 3893–3900.

Team G, Anil R, Borgeaud S, Wu Y, Alayrac J, Yu J, Soricut R, Schalkwyk J, Dai A, Hauth A et al. (2024) Gemini: A family of highly capable multimodal models, 2024. *arXiv preprint arXiv:2312.11805* .

Tian M, Giunchiglia F, Song R, Chen X and Xu H (2023) Enhancing ontology translation through cross-lingual agreement. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 1–5.

Toro S, Anagnostopoulos AV, Bello SM, Blumberg K, Cameron R, Carmody L, Diehl AD, Dooley DM, Duncan WD, Fey P et al. (2024) Dynamic retrieval augmented generation of ontologies using artificial intelligence (dragon-ai). *Journal of Biomedical Semantics* 15(1): 19.

Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F et al. (2023a) Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* .

Touvron H et al. (2023b) Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288* .

Treviso M, Lee JU, Ji T, Aken Bv, Cao Q, Ciosici MR, Hassid M, Heafield K, Hooker S, Raffel C et al. (2023) Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics* 11: 826–860.

Tsaneva S, Vasic S and Sabou M (2024) Llm-driven ontology evaluation: Verifying ontology restrictions with chatgpt. *The Semantic Web: ESWC Satellite Events* 2024.

Tufek N, Saissre A and Hanbury A (2024) Validating semantic artifacts with large language models. In: *Proceedings of the 21th European Semantic Web Conference (ESWC), Krete, Greece*. pp. 24–30.

Usmanova A and Usbeck R (2024) Structuring sustainability reports for environmental standards with LLMs guided by ontology. In: *Proceedings of the 1st Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2024)*. pp. 168–177.

Vaithilingam P, Zhang T and Glassman EL (2022) Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In: *Chi conference on human factors in computing systems extended abstracts*. pp. 1–7.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł and Polosukhin I (2017) Attention is all you need. *Advances in neural information processing systems* 30.

Vieira ES and Gomes JA (2009) A comparison of scopus and web of science for a typical university. *Scientometrics* 81: 587–600.

Volz R, Kleb J and Mueller W (2007) Towards ontology-based disambiguation of geographical identifiers. In: Bouquet P, Stoermer H, Tummarello G and Halpin H (eds.) *Proceedings of the WWW2007 Workshop I^3: Identity, Identifiers, Identification, Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8, 2007*, *CEUR Workshop Proceedings*, volume 249. CEUR-WS.org. URL https://ceur-ws.org/Vol-249/submission_132.pdf.

Vrandečić D and Krötzsch M (2014) Wikidata: a free collaborative knowledgebase. *Communications of the ACM* 57(10): 78–85. DOI:10.1145/2629489.

Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D et al. (2022) Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35: 24824–24837.

West P, Bhagavatula C, Hessel J, Hwang JD, Jiang L, Bras RL, Lu X, Welleck S and Choi Y (2021) Symbolic knowledge distillation: from general language models to commonsense models. *arXiv preprint arXiv:2110.07178* .

Wu H and Yu X (2024) The importance of modular structure in artificial intelligence algorithm evaluation. In: *2024 International Conference on Intelligent Education and Intelligent Research (IEIR)*. IEEE, pp. 1–6.

Wu J, Gan W, Chen Z, Wan S and Yu PS (2023) Multimodal large language models: A survey. In: *2023 IEEE International Conference on Big Data (BigData)*. IEEE, pp. 2247–2256.

Wu T, Luo L, Li YF, Pan S, Vu TT and Haffari G (2024) Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364* .

Xie Q, Luo Z, Wang B and Ananiadou S (2023) A survey for biomedical text summarization: From pre-trained to large language models. *arXiv preprint arXiv:2304.08763* .

Xu F, Lin Q, Han J, Zhao T, Liu J and Cambria E (2025) Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *IEEE Transactions on Knowledge and Data Engineering* .

Yang R, Tan TF, Lu W, Thirunavukarasu AJ, Ting DSW and Liu N (2023) Large language models in health care: Development, applications, and challenges. *Health Care Science* 2(4): 255–263.

Zamazal O (2024) Towards pattern-based complex ontology matching using sparql and llm. In: *Proceedings of the 20th International Conference on Semantic Systems (SEMANTiCS 2024), SEMANTiCS, Amsterdam, Netherlands*.

Zhang B, Carriero VA, Schreiberhuber K, Tsaneva S, González LS, Kim J and de Berardinis J (2025) Ontochat: A framework for conversational ontology engineering using language models. In: Meroño Peñuela A, Corcho O, Groth P, Simperl E, Tamma V, Nuzzolese AG, Poveda-Villalón M, Sabou M, Presutti V, Celino I, Revenko A, Raad J, Sartini B and Lisena P (eds.) *The Semantic Web: ESWC 2024 Satellite Events*. Cham: Springer Nature Switzerland, pp. 102–121.

Zhang D, Yu Y, Dong J, Li C, Su D, Chu C and Yu D (2024a) Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601* .

Zhang Y, Yang H, Wang H and Zhao J (2024b) Fast adaptation via prompted data: An efficient cross-domain fine-tuning method for large language models. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. pp. 7117–7132.

Zhang Z, Zhang A, Li M, Zhao H, Karypis G and Smola A (2023) Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923* .

Zhao H, Chen H, Yang F, Liu N, Deng H, Cai H, Wang S, Yin D and Du M (2024) Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology* 15(2): 1–38.