Large Language Models for Ontology Engineering: A Systematic Literature Review

Journal Title
XX(X):1–57

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 36 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 auxiliary ontology engineers, domain experts, and evaluators, using models such as GPT, LLaMA, and T5 models. Different approaches use zero, and few-shot prompt techniques 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

Prepared using sagej.cls *[Version: 2017/01/17 v1.20]*

1 Introduction

Ontologies have emerged as a crucial technology for providing machine-readable semantics and structured knowledge representations that enable data integration, validation, and automated reasoning over data (Patel and Debnath (2024); Glauer et al. (2024); Krötzsch and Thost (2016)). Ontologies are employed in a wide range of applications (ranging from IoT (Janowicz et al. (2019) or digital rights (Rodríguez-Doncel et al. (2018)) to Biology (Ashburner et al. (2000)) to define domain-specific concepts, relationships, constraints, axioms and logical rules (De Vergara et al. (2004); Patel and Debnath (2024); Glauer et al. (2024)).

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 requires deep domain expertise, careful conceptual modeling, extensive collaboration among stakeholders, and precise alignment with intended use cases.

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 need 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 OE. We extend the overview presented in our previous work (Garijo et al. (2024)) with the following contributions: First, through a comprehensive systematic search, we broaden and update the literature coverage, ultimately identifying 36 peer-reviewed papers published between 2018 and May 2025, significantly more than the 20+ papers included in the earlier overview. Second, we introduce a structured analytical framework that categorizes existing research according to ontology engineering stages, LLM roles, LLM technical details, and input/output formats for and from each LLM. Third, we examine dataset usage, evaluation practices, and the degree of human involvement in LLM-supported

workflows. Finally, we analyze the application domains in which LLMs have been deployed for ontology development, offering additional cross-study insights. Building on these contributions, this extended study aims to achieve 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. (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]. In addition, the corpus is archived on Zenodo (Li et al. (2025)).

The remainder of this article is organized as follows. Section 2 presents background information on OE and LLM technologies. Section 3 outlines our research objectives and key questions and describes the data collection and analysis methods. Section 4 presents the research results and key insights. Section 5 shows the discussion of the analyzed studies, and Section 6 concludes the survey by highlighting open research challenges. Finally, Section 8 describes the supporting materials used in our work.

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 the representation of structured knowledge (Dimitropoulos and Hatzilygeroudis (2024)) and facilitating semantic interoperability between systems and applications (Bittner et al. (2005); Tan et al. (2024)).

Ontology engineering (OE) provides the methodologies and tools necessary to construct domain-specific and application-specific ontological models (Gómez-Pérez (1999)). An ontology engineering method (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.

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

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 such as import, versioning, mapping, and modularization.

To consider a basic group of activities that are usually carried out during ontology development, we follow the Linked Open Terms (LOT) methodology (Poveda-Villalón et al. (2022)) general workflow as it includes the 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 more activities than the ones listed below; however, we include in this section only those activities found in the reviewed papers.

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

    - *Functional requirement 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 (CQs) Gruninger (1995) 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. (2024b)) and aligns with NeOn Ontological Resource Reverse Re-engineering (Suárez-Figueroa et al. (2012)).
    - *Requirement formalization:* This activity consists of 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:* Identify and report 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 that have demonstrated remarkable performance across tasks like text generation (Mishra et al. (2025); Wu (2024)), question answering (Arefeen et al. (2024); Balepur et al. (2025)), translation (Brown et al. (2020)), summarization (Azher et al. (2025)), and sentiment analysis (Kheiri and Karimi (2024)). 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. (1986) introduced recurrent neural networks (RNNs), which were later enhanced by the Long Short-Term Memory (LSTM) model developed by 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) demonstrated increasingly sophisticated generative capabilities (Brown et al. (2020); Radford et al. (2019)). GPT-3, for instance, was trained on 45TB of data and contained 175 billion parameters. In 2023, Meta introduced LLaMA, an open-source LLM trained on 1.4 trillion tokens across multiple model sizes (Raiaan et al. (2024)). Since then, models such as Google Gemini (Team et al. (2024)), OpenAI's GPT-4 (OpenAI et al. (2024)), Meta 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 (Wei et al. (2022)), code generation (Vaithilingam et al. (2022); Jiang et al. (2024)), and multimodal tasks (Zhang et al. (2023b); 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)).

Prompt engineering has emerged as a key methodology for enhancing the performance of pre-trained LLMs (Debnath et al. (2025)). It involves the careful design of instructions, conveyed through text, images, audio, or other modalities, that serve as the primary interface to guide LLMs in downstream tasks (Marvin et al. (2023)). A wide variety of prompting strategies have been developed to steer models toward accurate and contextually appropriate outputs. For example, zero-shot prompting is based exclusively on a task description, allowing models to generalize to unseen tasks without any examples (Radford et al. (2019)). In contrast, one-shot (Kojima et al. (2022)) and few-shot (Brown et al. (2020)) prompting incorporate one or several demonstrations, helping the model better infer input–output relationships (Kadam and Vaidya (2018)).

Other techniques aim to improve the structure and consistency of model outputs. Role prompting (Zheng et al. (2024); Olea et al. (2024)) assigns the model a specific persona or professional role, thus shaping its reasoning style and lexical choices. Template-based prompting (Shin et al. (2020)) employs predefined templates populated with task-specific variables to enforce structured formats such as JSON, tables, or logical expressions.

Chain-of-Thought (CoT) prompting (Wei et al. (2022)) augments few-shot learning by guiding models to articulate intermediate reasoning steps before delivering final answers. Also referred to as Chain-of-Thoughts in some studies (Besta et al. (2024); Chen et al. (2023)), this approach has been shown to substantially enhance LLM performance in mathematical and reasoning tasks. Typical CoT prompts include exemplar questions paired with reasoning traces and correct answers. The Reasoning and Acting (ReAct) framework (Yao et al. (2022)) extends CoT by interleaving reasoning with executable actions. When solving a problem, the model iteratively generates a thought, takes an action, and observes the outcome, maintaining a contextual memory by incorporating past reasoning steps, actions, and observations into the prompt. Further advancing multi-step reasoning, self-consistent sampling enhances output reliability by selecting the most consistent answer from multiple reasoning trajectories. Building on these foundations, frameworks such as ReAct (Yao et al. (2022)) and Tree-of-Thought (ToT) (Yao et al. (2023)) integrate systematic reasoning with action execution or structured search, supporting more sophisticated decision-making processes.

Fine-tuning is a process in which a pretrained model, such as an LLM, is further trained on a custom data set to adapt it for specialized tasks or domains. Complementing prompt-based approaches, fine-tuning provides a parameter-level adaptation mechanism that aligns LLMs with specific domains or tasks (Anisuzzaman et al. (2024)). Methods such as instruction tuning (Zhang et al. (2023a)), domain-specific fine-tuning (Gajulamandyam et al. (2025)), and parameter-efficient (Liu et al. (2022)) approaches such as LoRA (Hu et al. (2022)) or adapter tuning (Le et al. (2021)) enable models to internalize task patterns beyond the reach of prompts alone. Fine-tuning enhances output stability, mitigates prompt sensitivity, and ensures consistent performance in scenarios that require specialized knowledge or complex reasoning.

Overall, prompting techniques have evolved into a flexible and reusable interaction layer that complements advances in LLM architectures. By enabling more accurate, controllable, and domain-aligned outputs, prompt engineering has become central to the effective deployment of LLMs, serving as a key driver of innovation across AI applications.

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 LLM prompt techniques are employed to support OE activities (e.g.,zero-shot prompt, iterative prompt, fine-tuning)?

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

RQ2.e What outputs are generated by the LLMs?

RQ2.f 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 perform a systematic search in open-access digital libraries to ensure comprehensive coverage of the area under investigation (Vieira and Gomes (2009)). We selected Google Scholar, Web of Science and Scopus 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 depends on the following inclusion and exclusion criteria:

1. **Publication Time Frame:** We focus on papers published between 2018 and May 2025 to capture the most recent advances in ontology development driven by large language models (LLMs). The year 2018 marks a pivotal milestone in NLP, corresponding to the introduction of the Transformer architecture and the release of foundational models such as BERT (Devlin et al. (2019)) and GPT (Radford and Narasimhan (2018)), which laid the foundations for the modern LLM paradigm.

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

[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

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

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

   (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 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 paper 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` and archived on Zenodo (Li et al. (2025)). 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, technique of prompt 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 in which LLMs were applied, such as healthcare, education, and finance.

4 Search Results

Our search retrieved $\mathbf{15,688}$ records, with $\mathbf{7,179}$ unique papers after removing duplicates. We then conducted a manual screening in four stages. The first stage involved title screening to exclude clearly irrelevant studies (e.g., general LLM applications, safety/governance work, and ontology systems without explicit ontology engineering goals), resulting in $\mathbf{271}$ papers. The second stage was an abstract screening process, which further reduced the set to $\mathbf{70}$ papers by excluding studies that did not genuinely address LLM-based ontology development (e.g., those primarily focusing on ontology information extraction or ontology extension). The third stage included a peer-review status check, which yielded $\mathbf{46}$ eligible papers. Finally, in stage four, a full screening of the eligible papers was conducted. As a result, a total of $\mathbf{10}$ papers were excluded for final analysis: $\mathbf{2}$ papers discussed the requirements for benchmarks rather than introducing a new evaluation/method (Alharbi et al. (2024a); Plu et al. (2024)), $\mathbf{2}$ were review papers (Garijo et al. (2024); Perera and Liu (2024)), and $\mathbf{6}$ were domain studies primarily focused on extraction/population of knowledge graphs (Usmanova and Usbeck (2024); Mukanova et al. (2024); Sahbi et al. (2024); Tian et al. (2023); Funk et al. (2023); Straková et al. (2023)). Consequently, $\mathbf{36}$ papers were retained for the final analysis.

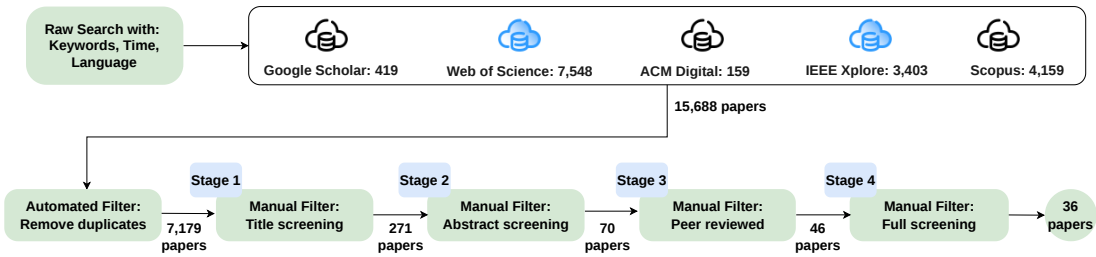

Figure 1. Paper selection process based on our methodology. From $\mathbf{15,688}$ papers retrieved across five libraries, $\mathbf{36}$ papers related to our LLM-based OE tasks were selected after applying an automated filter and a manual filter.

Figure 2 shows an overview of the reviewed works, grouping tasks into four OE phases (requirements specification, implementation, publication, and maintenance) to reflect the staged OE lifecycle. In total,

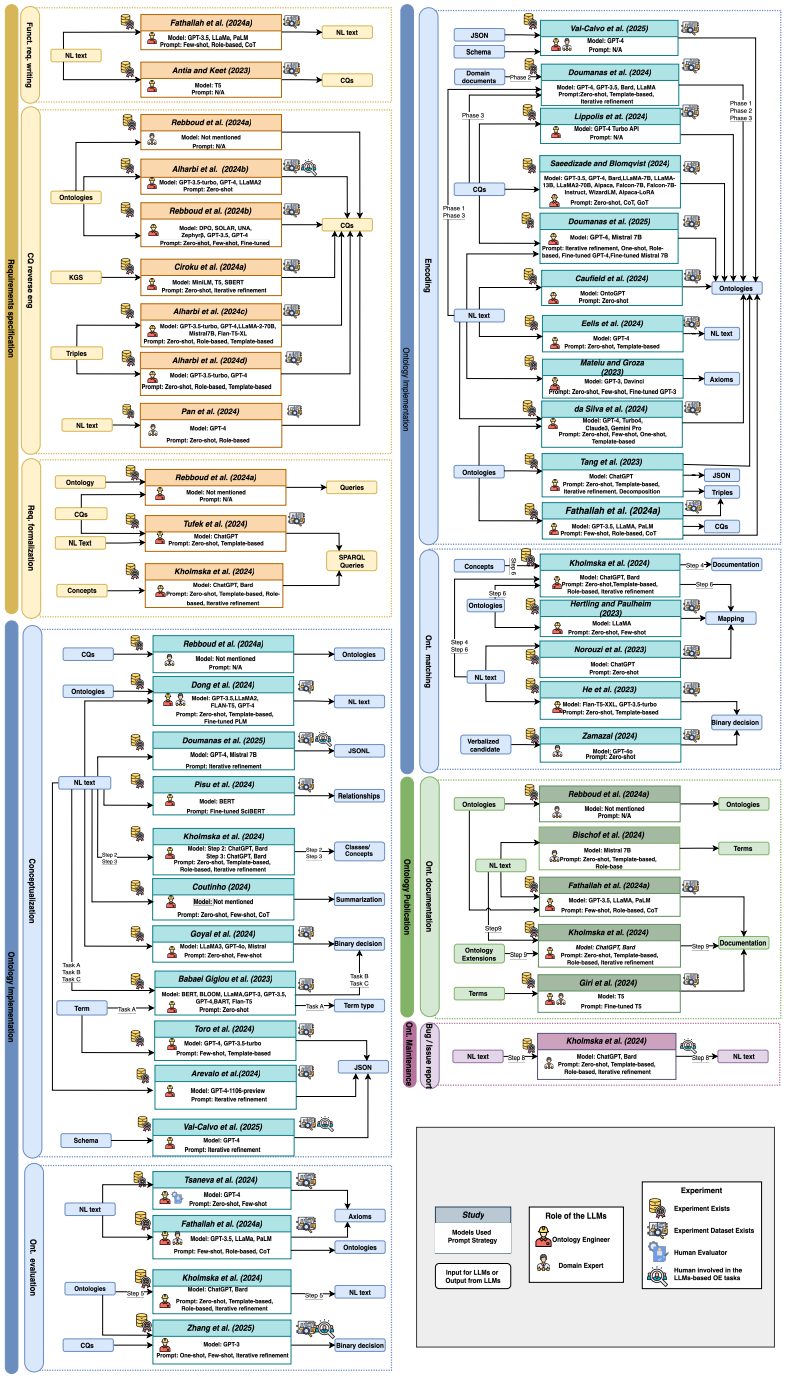

Figure 2. Overview of LLM-supported OE tasks based on 49 task-level studies from 36 papers.

Prepared using sagej.cls

our analysis covers **49** task-level studies extracted from the **36** included papers. It also summarizes the key dimensions examined in our research questions, including LLM inputs and outputs, model types, prompting strategies, the extent of human participation, LLM functional roles, and the availability of evaluation evidence and datasets. In the figure, rounded rectangles denote LLM inputs (e.g., natural language text (NL), CQs) and corresponding outputs, while icons indicate human involvement and whether experiments or datasets are reported. Task and step labels (e.g., Task A, Step 2) mark LLM-involved workflow steps within a task, including cases where a task is decomposed into sub-tasks or sequential steps. To ensure clear alignment with the research design described in Section 3.1, the findings are organized by research question: Section 4.1 addresses RQ1; Section 4.2 reports results for RQ2 (RQ2.a–RQ2.e); Section 4.3 presents findings for RQ3 (RQ3.a–RQ3.d); and Section 4.4 addresses RQ4.

4.1 RQ1:Overview of LLM-Supported Ontology Development Activities

The first step in our study is to analyze in which OE 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 3, 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 the OE activity addressed.

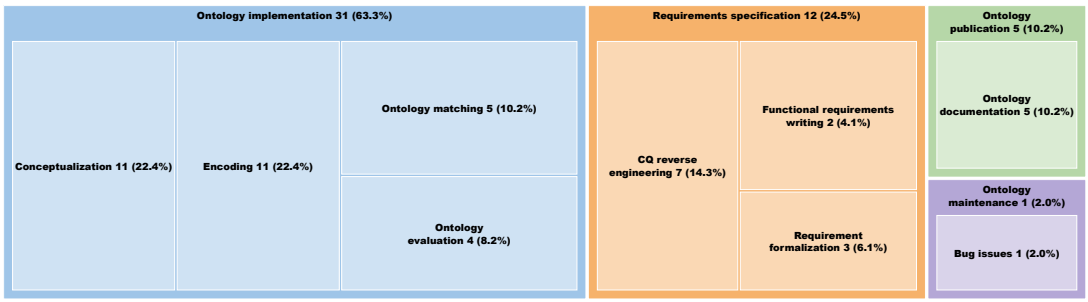

Figure 3. Distribution of LLM-supported tasks across ontology development phases based on 49 task-level studies from 36 papers. Numbers represent the total number of tasks identified for each phase, and percentages indicate their proportion relative to all tasks. Most tasks focus on ontology implementation (31 studies, 63.3%), followed by requirements specification (12 studies, 24.5%), publication (5 studies, 10.2%), and maintenance (1 study, 2.0%).

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 to generate ontology requirements from natural

[3] https://oeg-upm.github.io/llm4oe-slr/Figures/Taxonomic%20framework%20of%20LLM-supported%20ontology%20engineering%20tasks.svg

Table 1. Summary of ontology development phases, tasks, resources, inputs, and outputs supported by LLMs. For studies applying LLMs across multiple workflow stages (e.g., Doumanas et al. (2024); Kholmska et al. (2024)), we list each task separately to capture 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. (2024b)	Triples	CQs
		Ciroku et al. (2024a)	KGs	CQs
		Rebboud et al. (2024b)	Ontologies	CQs
		Alharbi et al. (2024c)	Triples	CQs
		Pan et al. (2024)	Natural language text	CQs
		Kholmska et al. (2024)	Concepts	SPARQL Queries
	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
		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
		Pisu et al. (2024)	Natural language text	Relationships
		Val-Calvo et al. (2025)	JSON	Schema
		Arevalo et al. (2024)	Natural language text	JSON
		Doumanas et al. (2025)	Natural language text	JSONL
	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
		Lippolis et al. (2024)	CQs	Ontologies
		Val-Calvo et al. (2025)	JSON and Schema	Ontologies
		Doumanas et al. (2025)	Natural language text, CQs	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	Bischof et al. (2024)	Natural language text	Terms
		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

language text 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. (2024b) proposed RETROFIT-CQs, which extracts RDF triples from existing ontologies and uses them to instantiate prompt templates for automatically generating candidate CQs. In the follow-up

work, Alharbi et al. (2024d) extended this CQ generation work, by systematically comparing prompt variants (P1–P3), ranging from a minimal baseline (P1), to progressively enriched prompts with additional guidance and CQ definitions (P2) and role-augmented prompts (P3). Rebboud et al. (2024a) introduced a benchmarking strategy that includes generating CQs from ontologies, using tools such as LangChain and Ollama.

Several additional contributions enrich this area. For example, Ciroku et al. (2024a) developed RevOnt, a system to extract 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. Pan et al. (2024) used a Retrieval Augmented Generation (RAG) approach (Arslan et al. (2024)) to generate CQs for two ontology engineering tasks, incorporating retrieved scientific passages as contextual input in the prompt for CQ-generation.

Once requirements and CQs are established, **requirement formalization** can automate the transfer of CQs into executable queries, a crucial step in ontology development. Kholmska et al. (2024) investigated the 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 such as Tree edit distance (TED (Pawlik and Augsten (2016)). 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 identifying the core terms, relationships, taxonomies, and early constraints/axioms that capture the intended semantics. Our analysis identified 11 task-level 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 to find the candidate concepts from domain sources. In addition, LLMs have also improved taxonomy discovery and relationship extraction. Studies by Goyal et al. (2024) and Babaei Giglou et al. (2023) employed LLMs to support ontology conceptualization through the identification of semantic relations, showing that LLMs can detect both taxonomic and non-taxonomic relationships between concepts.

Several researchers have also proposed integrated frameworks for the ontology conceptualization task. For instance, Coutinho (2024) proposed a system that integrated LLMs with textual ontology representations to generate candidate concepts from context, guided by the Unified Foundational Ontology (UFO)(Guizzardi et al. (2015)). Further contributions include Rebboud et al. (2024a), who framed this task as the construction of 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, organize them hierarchically, and produce definitions to support concept verification and refinement, demonstrating the potential of LLMs for concept discovery and structuring.

Dong et al. (2024) explored concept generation, while Toro et al. (2024) introduced techniques to complete ontology terms. Pisu et al. (2024) investigated the use of LLMs for the generation and construction of taxonomies of research topics. More recently, Val-Calvo et al. (2025) proposed a schema-level conceptualization step using an LLM-based agent to refine the high-level ontology structure into

detailed prompts that guide ontology building. In parallel, Arevalo et al. (2024) demonstrated automatic extraction of key concepts and relations to construct a domain ontology from NLP-focused OpenAlex[7] articles. Doumanas et al. (2025) extracted concepts and relations from domain texts to construct supervision data for fine-tuning LLMs, which were then used in their ontology generation pipeline.

Encoding refers to the translation of conceptual models into formal ontology representation languages. Our survey identified 11 task-level studies that investigated how LLMs can support this process. In the context of domain-specific formalization, Doumanas et al. (2024) and Doumanas et al. (2025) employed LLMs to develop an OWL ontology for search and rescue missions (SAR). Their evaluation was performed against the gold reference ontology in the SAR domain (Masa et al. (2022)).

Several studies focus on transforming natural language into OWL artifacts. Mateiu and Groza (2023) developed a Protégé plugin[8] that converts natural-language sentences into OWL axioms using LLMs. Similarly, Caufield et al. (2024) proposed a pipeline to extract procedural knowledge from web sources (e.g., recipes) and encode it into ontology structures. Eells et al. (2024) prompted LLMs to generate ontologies for common nouns and assessed syntactic validity and structural completeness. Saeedizade and Blomqvist (2024) investigated OWL generation from structured narratives, and Tang et al. (2023) demonstrated domain-specific encoding for road-traffic knowledge in autonomous-driving settings.

Recent work also highlights CQ/schema-driven encoding pipelines. Lippolis et al. (2024) used a gold-standard dataset of CQ–SPARQL-OWL pairs to support staged ontology refinement under the eXtreme Design methodology (Blomqvist et al. (2016)), where LLM interpreted CQs are used to derive classes, properties, and restrictions. Complementarily, Val-Calvo et al. (2025) introduced a modular workflow in which an *Ontology Building module* generates a final ontology from the structure defined in earlier schema-design stages.

In addition, da Silva et al. (2024) proposed an LLM-based method for transforming capability descriptions into ontological models, reducing manual effort in ontology creation from natural language inputs. Fathallah et al. (2024a) further presented a pipeline that automates ontology encoding by generating structured triples and ontology artifacts from textual inputs.

LLMs have also been applied to **ontology matching** tasks, which are key to ensuring 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 uses LLMs to generate high precision ontology mappings. Additionally, studies by Norouzi et al. (2023) and He et al. (2023) benchmarked the performance of LLMs in ontology alignment against reference mappings, revealing that modern LLMs can perform comparable to specialized alignment systems. Kholmska et al. (2024) further explored LLMs in generating initial mapping suggestions to support ontology extension, assessing concept coverage and inter-model consistency.

In the **ontology evaluation** task, LLMs have been applied 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 problems. Fathallah et al. (2024a) explored a different approach, proposing an evaluation framework that leverages ChatGPT in ontology syntax correction, using parsing errors detected by RDFLib and pitfall descriptions from the

[7] https://openalex.org/
[8] https://protege.stanford.edu/

OOPS! API (Poveda-Villalón et al. (2014)), particularly focusing on the 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. Kholmska et al. (2024) provided a broader evaluation of ontology quality, focusing on relevance, consistency of content, and structural soundness in the development of LLM-supported ontology.

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 5 task-level studies applying LLMs to ontology documentation tasks.

Bischof et al. (2024) employed LLMs to produce context-sensitive annotations aligned with domain-specific conventions. Rebboud et al. (2024a) explored the use of LLM to generate structured documentation of key ontology components, such as classes and properties; their evaluation, based on semantic similarity metrics, showed that the LLM-generated documentation is accurate and relevant. Fathallah et al. (2024a) further addressed natural language generation for ontology entities and properties, enhancing comprehensibility for both technical and non-technical users. In more specialized domains, Giri et al. (2024) applied the T5 language model to summarize functional descriptions of Gene Ontology terms, while Kholmska et al. (2024) leveraged LLMs to create comprehensive documentation for extended ontologies, supporting knowledge sharing and reuse.

4.1.4 Ontology maintenance

Among the studies reviewed, only one 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 LLMs can effectively support ontology maintenance.

4.1.5 Summary

Based on the analysis of 49 OE task-level studies from 36 papers, LLMs have been applied unevenly in different ontology development phases. The implementation phase dominates, with 31 studies focused on conceptualization, encoding, matching, and evaluation. Requirements specification ranks second, represented by 12 studies addressing functional requirements, competency question generation, and formalization into SPARQL queries. Later stages receive limited attention: 5 studies focus on ontology publication through documentation generation, while only 1 addresses maintenance tasks.

4.2 RQ2:LLM-based Approaches to Ontology Development

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, refer to Table 2 in the Appendix.

4.2.1 RQ2.a: Role of LLMs in OE activities

Based on the reviewed studies, LLMs take on several key collaborative roles within OE tasks, either complementing or, in some cases, replicating tasks traditionally performed by human knowledge engineers. These contributions can be grouped into four main categories:

1. **Ontology Engineer**: LLMs are increasingly functioning as automated Ontology Engineers, actively supporting the design, development, and maintenance of ontologies throughout the entire development lifecycle. More precisely, LLMs are utilized to **(a)** parse unstructured domain texts and generate structured requirement specifications, thereby facilitating automated requirement elicitation (Alharbi et al. (2024b,d); Ciroku et al. (2024a)); **(b)** transform competency questions into structured queries (e.g., SPARQL) (Rebboud et al. (2024a); Tufek et al. (2024)); **(c)** discover axioms, particularly identifying 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 directly into OWL code (Doumanas et al. (2024); Eells et al. (2024); Saeedizade and Blomqvist (2024); Tang et al. (2023); Lippolis et al. (2024); Doumanas et al. (2025)); and **(e)** support the entire ontology lifecycle, from conceptualization through to documentation, or provide end-to-end assistance under methodologies such as the NeOn-GPT approach (Kholmska et al. (2024); Fathallah et al. (2024a)).

2. **Domain Experts**: LLMs act as domain experts by supporting knowledge extraction, term definition, and ontology content validation. They perform tasks requiring domain-specific understanding, such as **(a)** generating domain-relevant concepts (Dong et al. (2024)); **(b)** producing context-sensitive term annotations (Bischof et al. (2024)); **(c)** generating structured ontology documentation (Rebboud et al. (2024a); Giri et al. (2024)); and **(d)** summarizing functional descriptions (Consortium (2006)). They are also used in evaluation tasks requiring both technical and domain expertise to assess the consistency and correctness of ontology content (Tsaneva et al. (2024); Fathallah et al. (2024a)). **(e)** Additionally, LLMs assist in validating or suggesting domain-specific relations and constraints, ensuring alignment with established domain semantics (Babaei Giglou et al. (2023); Goyal et al. (2024)).

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: 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, due to their strong capabilities in natural language understanding and generation (Rebboud et al. (2024b); Tufek et al. (2024); Fathallah et al. (2024a); Arevalo et al. (2024); Val-Calvo et al. (2025)). 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, etc.):** Open source LLMs such as LLaMA (Touvron et al. (2023a)), Mistral[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-GPT framework proposed by Fathallah et al. (2024a), which supported 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. (2024c) employed models such as LLaMA-2-70B, Mistral 7B (Jiang et al. (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 ability of these models to process narrative ontology descriptions and associated CQs for automated ontology creation, compared to 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 to 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: LLMs prompt techniques employed to support OE activities

To understand how LLMs are operationalized within OE workflows, this subsection examines the prompting techniques used in the reviewed studies. The analysis covers the full range of strategies used to guide or adapt LLM behavior, including zero-shot and few-shot prompting, role-based prompting, template-based and representational prompting, reasoning-driven prompting, iterative refinement, retrieval-augmented prompting, and fine-tuning.

[9] https://mistral.ai/

- **Zero-shot prompting:** Zero-shot prompting is the most frequently used strategy and is applied when tasks can be specified purely through natural-language instructions. It is used across requirements specification, conceptualization, encoding, and ontology matching. In many cases, zero-shot prompting is combined with structural templates to constrain output formats. Examples include CQs generation from textual descriptions or triples (Rebboud et al. (2024b); Alharbi et al. (2024b)), SPARQL query generation using instruction-only templates (Tufek et al. (2024)), and type classification using only local context (Goyal et al. (2024)). In ontology encoding, natural language definitions are translated into OWL axioms via zero-shot templates specifying the expected syntax Caufield et al. (2024). Alignment approaches relying solely on verbalized labels also follow a pure zero-shot setup (He et al. (2023); Norouzi et al. (2023)). Zero-shot prompting is further used for capability modeling based on TBox grounding (da Silva et al. (2024)).

- **Few-shot and one-shot prompting:** Few-shot prompting augments instructions with a small number of examples, improving structural fidelity and reducing hallucinations. One-shot prompting provides a single demonstration when minimal scaffolding suffices. These techniques are widely used for CQ generation, conceptualization, evaluation, and matching. Few-shot examples improve CQ extraction (Rebboud et al. (2024b)), guide entity–relation extraction in NeOn-GPT (Fathallah et al. (2024a)), and support axiom evaluation and alignment (Tsaneva et al. (2024); Hertling and Paulheim (2023)). One-shot prompting is used to illustrate user-state structures (Zhang et al. (2025)) or capability modeling patterns (da Silva et al. (2024)). In several workflows, example-driven prompting interacts with decomposition strategies.

- **Template-based prompting:** Template-based prompting uses fixed syntactic or structural scaffolds such as JSON schemas, CQ templates, SPARQL skeletons, or OWL functional syntax to constrain and standardize model outputs. This technique often appears in combination with zero-shot or few-shot prompting. Documentation templates specify fields such as labels and definitions (Bischof et al. (2024)). Triple-based templates standardize CQ phrasing (Alharbi et al. (2024b,d,b)). Encoding pipelines frequently use JSON schemas specifying IRIs, definitions, and axioms (Toro et al. (2024)). SPARQL templates enforce good form and reduce ambiguity (Tufek et al. (2024)). Many workflows combine templates with iterative correction loops. Tsaneva et al. (2024) showed that providing axioms in the Rector or Turtle format improves the verification accuracy. Verbalized labels, definitions, and structural fragments are also used in matching workflows (Hertling and Paulheim (2023)).

- **Role-based prompting:** Role-based prompting frames the model as an "ontology engineer", "domain expert", or "SPARQL specialist", grounding instructions in domain expertise. This technique often appears in combination with zero-shot, few-shot, CoT, or template-based prompting. Role prompts are used in requirement specification (Alharbi et al. (2024c); Pan et al. (2024); Alharbi et al. (2024d,b)), SPARQL generation (Kholmska et al. (2024)), conceptualization (Fathallah et al. (2024a)), documentation (Bischof et al. (2024)), ontology matching (Kholmska et al. (2024)), and evaluation (Fathallah et al. (2024a)).

- **Multi-step reasoning prompting (CoT, GoT, decomposition):** Reasoning-oriented prompting guides models through intermediate steps or decomposed subtasks. Chain-of-thought prompting

supports term extraction, classification, and axiom justification (Fathallah et al. (2024a)). Graph-of-Thoughts prompting enables multi-branch exploration of ontology structures (Saeedizade and Blomqvist (2024)). Decomposition strategies, often combined with templates or examples, break workflows into sequential steps (e.g., concepts → definitions → properties → axioms) as in ontology encoding (Tang et al. (2023)). Multi-step reasoning is also used in the validation of axioms (Tsaneva et al. (2024)).

- **Iterative and conversational refinement prompting:** Many OE workflows employ multi-turn refinement, in which model outputs are progressively revised based on constraints or feedback. In the conceptualization stage, recent studies use iterative refinement to improve high-level ontology design. For example, OntoGenix (Val-Calvo et al. (2025)) followed a human-agent loop in which an agent generates schema-level prompts (e.g., a *Prompt Crafter*), another agent plans revisions (e.g., a *Plan Sage*), and the prompts are iteratively updated based on this feedback before building the ontology. Similarly, AutOnto (Arevalo et al. (2024)) applied iterative prompt refinement to derive concepts and relations from NLP corpora before ontology construction.

  Iterative refinement is also widely used in encoding-related tasks, including multi-phase encoding pipelines (Doumanas et al. (2024)), multi-turn axiom correction (Fathallah et al. (2024a)), and SAR ontology engineering workflows. RevOnt (Ciroku et al. (2024a)) further implemented staged refinement for verbalization, abstraction, generalization, and CQ filtering. In addition, conversational refinement systems enable users to interactively adjust the generated CQs or alignments (Zhang et al. (2025)).

- **Fine-tuning and model adaptation:** A small subset of studies use supervised fine-tuning to adapt models to OE tasks. GPT-4 and Mistral-7B were trained on OE-specific JSONL datasets to support ontology generation tasks (Doumanas et al. (2025)). GPT-3 has been adapted for Natural Language (NL)-to-OWL translation tasks through task-specific prompting or tuning (Mateiu and Groza (2023)). T5 is fine-tuned on GO annotations for documentation (Giri et al. (2024)). The placement of Domain-specific concepts is enhanced through a fine-tuned BERT cross-encoder (Dong et al. (2024)). SciBERT is fine-tuned for the extraction of scientific relationships (Pisu et al. (2024)). In Rebboud et al. (2024b), most models are used in fine-tuned configurations, including FusionNet_7Bx2_MoE_14B SOLAR-10.7B-Instruct-v1.0, Mistral-7B and another Mistral-7B-v0.1. Although rare in general, fine-tuning yields notable gains for tasks requiring high structural or domain precision.

4.2.4 RQ2.d: Inputs For LLMs and Outputs from LLMs

To better analyze how LLMs are used across the OE lifecycle, we examine the inputs provided to LLMs and the outputs they generate in relation to the specific OE tasks they support. In this section, we summarize the recurring input–output patterns reported across the reviewed studies. We structure the analysis according to the OE activities identified in 4.1, as these activities naturally determine the expected output types.

- During the **ontology requirement specification** phase, common patterns are 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. (2024b); Ciroku et al. (2024b); Rebboud et al. (2024b); Alharbi et al. (2024c)); and (c) taking ontologies and natural language (including CQs) to generate queries as part of the requirement formalization activity (Tufek et al. (2024); Rebboud et al. (2024a)).

- 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 generated machine-readable representations, such as ontologies in OWL (Rebboud et al. (2024a)) or structured schemas in JSON (Toro et al. (2024); Arevalo et al. (2024); Val-Calvo et al. (2025)). Others produced concepts or terms intended for ontology integration (Kholmska et al. (2024); Babaei Giglou et al. (2023)). Also, some approaches generated 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 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)) took natural-language descriptions as input to generate ontology artifacts in OWL, RDF, or related formats. Two exceptions were Saeedizade and Blomqvist (2024) and Lippolis et al. (2024), which used CQs as input to guide ontology generation in alignment with user information needs. Regarding outputs, most approaches produced complete ontology code, with exceptions such as Mateiu and Groza (2023), which focused specifically on generating OWL axioms. In Eells et al. (2024), the LLM was prompted with a single noun (e.g., “air,” “book”) and returned a mix of natural language text and RDF ontology content.
  - To address **ontology matching**, some of the analyzed works took natural language inputs to produce binary decisions indicating semantic alignment. For example, Zamazal (2024) used LLMs to classify verbalized complex correspondence candidates as (probably) positive or negative, while He et al. (2023) evaluated 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) took 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) approached ontology reuse through a multi-step process: Step 4 employed LLMs to extract key features, such as purpose, reused elements, and formats from existing ontologies to support reuse decisions; Step 6 involved using LLMs to map new concepts to the previously identified ontologies by analyzing their definitions, relationships, and properties. The output

consisted of explicit mappings expressed as `owl:sameAs`, `owl:equivalentClass`, and `owl:equivalentProperty` statements. Similarly, Hertling and Paulheim (2023) combined 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 may 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 took 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) emphasized the production of readable summaries highlighting key classes and properties. Giri et al. (2024) generated concise summaries from Gene Ontology terms. Finally, Kholmska et al. (2024) leveraged LLMs to assist in the writing of technical reports.

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

4.2.5 RQ2.e: Role of humans in OE LLMs-assisted activities

Although many recent studies automate ontology development with LLMs, 5 studies explicitly involve human participants, typically domain experts or ontology engineers, to support tasks requiring judgment, contextual understanding, and refinement.

In conceptualization, Val-Calvo et al. (2025) introduced an explicit human-agent loop in which ontology engineers and domain experts iteratively refine schema-design prompts before ontology building. Complementarily, Doumanas et al. (2025) highlighted the importance of human oversight during training data preparation, where extracted outputs are validated and corrected to ensure that the resulting supervision data accurately reflects the intended knowledge and excludes errors or irrelevant content. Together, these studies demonstrated how human critique can guide LLMs toward more suitable high-level ontology structures. Doumanas et al. (2024) emphasized the crucial role of domain experts during the ontology

formalization/encoding. In particular, experts compare and evaluate the LLM-generated ontology against existing ontologies and, based on this human-driven and LLM-driven evaluation, propose a new ontology by combining existing and LLM-generated semantics. Similarly, Kholmska et al. (2024) described the involvement of domain experts and end-users during ontology maintenance and bug resolution. Their iterative feedback on errors and inconsistencies was critical to refine the ontology structure and enhance overall quality. In the context of ontology evaluation, Zhang et al. (2025) demonstrated how ontology engineers curated user stories that were manually authored or derived from earlier development stages to support meaningful CQ extraction, emphasizing the need for human input to link technical outputs to real-world use cases. Finally, Alharbi et al. (2024b) reported 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.

4.2.6 Summary

Across the 49 reviewed task-level studies, LLMs assumed three functional roles in OE: **Ontology Engineer**, **Domain Expert**, and **Human Evaluator**. Most studies relied on general-purpose models such as GPT-3.5 and GPT-4, with growing adoption of open-source models including LLaMA, Mistral, and Falcon. Current workflows predominantly use LLMs to automate the generation, transformation, and verification of ontology artifacts, while human involvement remains limited. Building on these model choices, the studies applied a wide range of prompting strategies. **Zero-shot** and **few-shot** prompting were the most common, while **template-based** prompts were adopted when outputs needed to follow predefined schemas. **Role-based prompting** contextualized instructions, and **multi-step reasoning** (e.g., CoT, GoT, decomposition), together with **iterative refinement**, supported staged modeling and corrective workflows. Fine-tuning was used only in a minority of studies requiring domain adaptation or high syntactic precision. Across these prompting settings, inputs ranged from unstructured text and competency questions to structured triples, knowledge graph fragments, and ontology modules. Outputs included classes, properties, OWL axioms, SPARQL queries, CQs, natural language definitions, and documentation. Only a small subset of studies involved explicit human feedback, typically through expert validation or interactive refinement.

4.3 RQ3: Evaluation of LLMs Performance in Ontology Development

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 8 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 Existence of Experiments

In this subsection, we examined whether the reviewed studies reported experimental evidence. Of the 49

OE task-level studies, 14 reported no explicit experimental evaluation, whereas the remaining 35 included experiments. Within these 35 task-level studies, Fathallah et al. (2024a) contributed four task-level entries (covering four OE activities) but reported only isolated LLM tests without baselines or comparative analysis. A further three studies reported results but did not provide publicly accessible resource links (Tsaneva et al. (2024); Alharbi et al. (2024c); Norouzi et al. (2023)). Overall, 32 studies provided accessible experiments with explicit evaluation metrics and comparative analysis.

Note that some studies appeared several times in our task-level analysis because they addressed multiple OE tasks (e.g., Fathallah et al. (2024a); Kholmska et al. (2024); Val-Calvo et al. (2025)), spanning requirements specification, conceptualization, encoding, and bug-related issues.

4.3.2 RQ3.b Datasets used

All studies except one Bischof et al. (2024) reported the datasets used for training, experiments, or evaluation. Across the remaining **48** task-level studies, datasets are mainly drawn from established ontology repositories and benchmark suites, relying primarily on structured ontology artifacts (OWL/RDF) and curated benchmarks (e.g., OAEI 2022 and the LLMs4OL Challenge), with some use of unstructured text corpora. Several ontologies (e.g., Dem@Care, SNOMED CT, and the Wine Ontology) are reused across multiple studies, suggesting that they function as de facto community benchmarks.

To examine the detailed datasets used, Alharbi et al. (2024c) selected four ontologies along with their associated CQ datasets to investigate CQ creation. Three of these ontologies: Video Game (entertainment) (Parkkila et al. (2017)), Dem@care (healthcare) (Karakostas et al. (2016)), and VICINITY Core (Internet of Things) (Cimmino et al. (2019)) were obtained from the CORAL (Fernández-Izquierdo et al. (2019)) repository, a comprehensive source for CQs. The fourth ontology, African Wildlife (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 base. 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. Tsaneva et al. (2024) used Pizza Ontology related from food domain 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 provided ontologies and KGs across various domains. Similarly, Babaei Giglou et al. (2023) and Goyal et al. (2024) adopted the LLMs4OL Challenge benchmark dataset, designed to assess LLM in various ontology learning tasks. This challenge spaned 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 facilitated the consistent evaluation of LLM-based methods in structured knowledge engineering.

In addition, several datasets have been reused in studies to allow a consistent evaluation of 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)) in 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 text as experimental input. 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 requirements specification task studies, Antia and Keet (2023) fed COVID-19 scientific papers into an automated CQ reverse engineering pipeline to derive candidate queries for ontology validation. Similarly, Pan et al. (2024) generated CQs for two tasks (KG-EmpiRE (Karras (2024)) and Human–Computer Interaction (HCI) (Costa et al. (2022)) by retrieving relevant evidence from the corresponding scientific corpora and injecting it into the CQ-generation prompts.

Beyond CQ generation, unstructured corpora have been leveraged for ontology construction. Arevalo et al. (2024) used natural-language text corpora together with the NLP subset of the CSO ontology (covering 156 deduplicated topics) to generate an ontology. In a commonsense setting, Eells et al. (2024) prompted LLMs with 101 high-frequency nouns from the Corpus of Contemporary American English (COCA) (Davies (2010)) to induce ontological structures, which were then assessed for semantic coherence and alignment with human commonsense knowledge.

To support further exploration of datasets used in LLM-based ontology engineering tasks, we provide Table 5 in the Appendix. The table lists acronyms and the full name of the datasets, the official or commonly used access link, and their associated domain, helping readers identify suitable datasets for specific domain applications.

4.3.3 RQ3.c: Evaluation Methods

In this section, we summarize the evaluation methods employed in the reviewed studies, as they are crucial for assessing the performance of LLM-driven OE activities. Specifically, we categorize evaluation designs as quantitative, qualitative, and mixed (hybrid). Quantitative evaluations typically compare model outputs against reference standards and report task-specific metrics (e.g., Precision/Recall/F1, BLEU, cosine

[10] https://schema.org

similarity). Qualitative evaluations rely on human judgment (e.g., expert review or manual inspection). Mixed (hybrid) designs combine automated metrics with human assessment.

For this subsection, we consider only 39 studies that specify an explicit evaluation protocol (e.g., clearly defined metrics, baselines, or comparison criteria), 10 studies without defined metrics/baselines or experiments are not included (Fathallah et al. (2024a); Tang et al. (2023); Mateiu and Groza (2023); Kholmska et al. (2024); Bischof et al. (2024)). Taking into account the remaining studies, three main evaluation approaches emerge, 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) applied the F1 score to measure the accuracy of LLM-generated output in ontology conceptualization tasks, as part of the LLMs4OL challenge. Alharbi et al. (2024c) and Kholmska et al. (2024) reported task-specific metrics such as consistency between models, reduction of errors, and coverage of concepts to assess the quality of generated ontologies. Dong et al. (2024) evaluated the predictions of hierarchical relationships using the insert rate at top k (InR@k), which reflected the precision with which new concepts are inserted into a taxonomy. Tufek et al. (2024) measured the accuracy of the exact match for the generation of SPARQL queries by comparing the outputs with predefined targets.
  - **Similarity-based evaluation:** Some studies applied semantic similarity measures, such as SentenceBERT cosine similarity, to compare LLM-generated outputs with reference texts, reducing the need for manual comparisons. Rebboud et al. (2024b) used SentenceBERT cosine similarity to evaluate the semantic relationship between LLM-generated competency questions and expert references. In a related setting, Rebboud et al. (2024a) proposed cosine similarity to compare the generated ontology documentation with expert definitions, supporting an efficient and consistent quality assessment.
  - **Ground truth-based evaluation:** Structural fidelity is evaluated using metrics such as 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. Although 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) incorporated 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 adopt a hybrid evaluation strategy that integrates both quantitative and qualitative methods. By combining metric-based assessments with expert reviews, these approaches validate both the structural quality and practical usability of LLM outputs, thereby enhancing evaluation robustness.

  In the context of verification and constraint checking, Tsaneva et al. (2024) compared the evaluation results generated by LLM with the majority vote of human experts to assess the feasibility and reliability of automated ontology restriction checking. da Silva et al. (2024) paired SHACL-based syntax validation with expert review to ensure logical consistency and eliminate redundancy in generated ontologies. Giri et al. (2024) incorporated human evaluation to validate the embedding-based confidence scores used in assessing LLM-generated biomedical summaries, examining how well automated scores align with expert ratings when high-confidence embeddings are observed.

  For ontology quality assessment, Lippolis et al. (2024) evaluated the LLM-based ontologies using standard quality metrics supplemented by domain expert review. Val-Calvo et al. (2025) combined automated quality analysis with expert-based manual assessment, employing OQuaRE metrics and the OOPS! pitfall scanner (Poveda-Villalón et al. (2014)) to quantitatively compare human-created and LLM-based ontologies, while ontology engineers qualitatively evaluate domain-relevant class representation. Arevalo et al. (2024) measured completeness and conciseness through pairwise average similarity, mean aggregate similarity, and cosine similarity against reference embeddings, complemented by qualitative assessment of accuracy, clarity, adaptability, and consistency via inspection of generated classes, properties, and relations.

  For broader evaluation, Coutinho (2024) integrated quantitative indicators such as task completion time and model quality metrics with qualitative insights from expert interviews and user satisfaction assessments, balancing automation with human feedback to improve inter-model consistency and overall usability. Alharbi et al. (2024b) similarly applied both paradigms. Quantitatively, they use metrics such as mean questions per triple, precision, recall, and F1-score to evaluate generated CQs. Qualitatively, they interviewed ontology developers to assess the intent and relevance of generated CQs, and invited ontology editors to rate predicted versus curated definitions.

4.3.4 RQ3.d: Performance Results

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.

In the requirements specification phase, multiple studies report that LLMs can effectively support CQ generation. Antia and Keet (2023) proposed AgOCQs, which feed COVID-19 scientific papers into an automated CQ generation pipeline to derive candidate questions for ontology validation. In a survey of 20 generated CQs, 70% were rated grammatically correct by at least 70% of participants. Ontology experts deemed 12/20 CQs answerable (50%–85% agreement across questions) and highly relevant (70%–93%), while 73% of users and 69% of experts agreed that the CQs provided clear domain coverage.

For CQ reverse engineering, Rebboud et al. (2024b) evaluated precision by matching generated CQs to gold CQs using SentenceBERT cosine similarity with a fixed threshold ($\theta = 0.6$). Among the evaluated LLMs, the open-source Mistral 7B models *Zephyr* β and *UNA* achieve strong best-case precision: *Zephyr* β reaches 0.90 on Odeuropa, while *UNA* reaches 0.31 on NORIA-O with few-shot prompting. Although *UNA* is not top-performing overall, it shows comparatively larger gains with few-shot prompting and remains competitive in schema-limited settings (i.e., when only classes and/or properties are provided rather than the full schema). Pan et al. (2024) further showed that a RAG approach improved CQ generation for the concrete *KG-EmpiRE* task, i.e., constructing a knowledge graph of empirical research in Requirements Engineering to capture its state and evolution, achieving a precision of ≈ 0.36 (vs. ≈ 0.05 for zero-shot) using a single carefully selected "visionary" paper as the knowledge base. The RevOnt framework (Ciroku et al. (2024a)) achieved strong performance, with a BLEU score of 0.41 in verbalization and 0.30 in question generation. More than 75% of its outputs were rated. In a complementary line of work, RETROFIT-CQ (Alharbi et al. (2024b)) adapted CQs to existing ontologies by generating questions from RDF triples. More than 75% of the generated CQs were directly executable as SPARQL queries without manual revision, indicating high structural compatibility with the target ontologies. Furthermore, Alharbi et al. (2024d) extended RETROFIT-CQ by enriching the prompt and by comparing two creativity settings, denoted as CP (CP=0.0: deterministic decoding; CP=0.7: default). Precision improved in some settings, for example, on the Video Game ontology with gpt-3.5-turbo under deterministic decoding (CP=0.0), precision increased from 45.2% with(P1) to 84.4%(P3).

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. The execution modality also mattered: the use of 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)). Arevalo et al. (2024) proposed AutOnto to derive a compact set of domain topics from text and evaluate it against a CSO subset using pairwise average similarity (PAS), mean aggregate similarity (MAS), and cosine similarity with reference embedding (CSRE). On the NLP domain, AO-NLP scores 0.34 in PAS (vs. 0.35 for CSO-NLP) and 0.84 in MAS (vs. 0.85 for CSO-NLP), and CSRE is 0.84, close to 0.88 from CSO-NLP, while using far fewer topics (56 vs. 156 deduplicated topics).

During encoding, Val-Calvo et al. (2025) compared OntoGenix-generated ontologies with human-developed ontologies in six datasets using OQuaRE (Duque-Ramos et al. (2011)) quality scores and OOPS! pitfalls and measure effort savings. OntoGenix yields time savings of 8.2%–58.3% for ontology development. In the SPIRES framework, GPT-3.5-turbo achieves perfect entity alignment; however, on the zero-shot chemical–disease relation task, SPIRES showed an F1-score of 43.8% (Caufield et al. (2024)). Moreover, da Silva et al. (2024) reported a mean error score of 0.03 for Claude and 0.12 for GPT under few-shot prompting, and completeness values of 0.90–1.00 for complex capabilities in few-shot settings.

In SAR ontology engineering, Doumanas et al. (2024) reported the highest F1-score under the X-HCOME evaluation setting for Bard at 48.21%, with precision 84% and recall 34.50%. Under the same setting, GPT-4 reached an F1-score of 30.92%, with a precision of 88% and a recall of 18.75%. They also reported that, when evaluated against a reduced gold-standard class hierarchy, the recall increases by 140% for GPT-4. In addition, Doumanas et al. (2025) fine-tuned GPT-4 and Mistral 7B using OE-specific JSONL datasets curated from foundational OE textbooks and re-evaluated SAR ontology generation against a human-expert reference ontology; for class generation, successive fine-tuning iterations improved GPT-4's F1-score from 19.35% to 27.08%.

In ontology matching and reuse, GPT-4o correctly validated complex alignments with 100% accuracy in rejecting false correspondences (Zamazal (2024)). The OLaLa study showed that improvements in the F1 score can achieve 90.2% with Llama-2-70b- instruct-v2, optimized for efficiency (Hertling and Paulheim (2023)). In the NCIT-DOID equivalence matching benchmarks (He et al. (2023)), Flan-T5-XXL achieved the highest F1-score, reaching 72.1% with a threshold of 0.650, and also achieved the best Hits@1 of 88.0%. Norouzi et al. (2023) reported precision 37%, recall 92%, and F1-score 52% on the OAEI 2022 benchmark, with the highest recall and F1-score obtained using the iterative prompt design that queries matches per class and property.

In ontology evaluation, ChatGPT-4 verified axioms with 92.2% accuracy, which increased to 96.7% using ensemble aggregation (Tsaneva et al. (2024)). OntoChat achieved 87.5% positive expert ratings for clustering competency questions (Zhang et al. (2025)). DRAGON-AI reported high precision but moderate recall, and its performance improved iteratively with user input (Toro et al. (2024)). In a controlled educational setting, GPT-4 with CQ-by-CQ prompting achieved CQ pass rates of up to 100% across several CQ categories when evaluated by executable SPARQL query success (Saeedizade and Blomqvist (2024)).

Finally, in the ontology maintenance task, GO2Sum (Giri et al. (2024)) outperformed vanilla T5 for 95.5%–98.0% of targets under embedding-based similarity and for 95.3%–97.3% under mover-based similarity metrics. In predicted GO annotations, 73.7%, 79.9%, and 95.7% of the summaries for Function, Subunit Structure, and Pathway, respectively, achieved an average embedding score greater than or equal to 0.5. These results show that LLM-based summarization improves semantic alignment with reference descriptions for low-coverage GO predictions, indicating the effectiveness of LLMs in supporting ontology debugging and interpretation.

4.3.5 Summary

Evaluation practices in LLM-based ontology engineering mainly adopt quantitative metrics such as precision, recall, F1-score, and semantic similarity. Several studies combine these with qualitative expert reviews to assess conceptual validity and domain relevance. Most evaluations focus on overall system output rather than isolating LLM performance, often using existing ontologies (e.g., SNOMED CT, FIBO) as benchmarks. Open-source datasets are increasingly used to improve reproducibility, while standardized protocols remain scarce. Several emerging initiatives, such as OAEI and LLMs4OL, have begun to define shared datasets and metrics. In general, evaluation remains fragmented in all studies, lacking unified criteria and alignment of standards.

4.4 RQ4: Application Domains of LLM-based Ontology Development

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 the development of domain-specific ontologies such as 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 were 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 the *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 helped 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 supported 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 highlighted the versatility of LLMs across diverse ontology-driven domains (see Table 4 in the Appendix for details). Most studies focused on life sciences and healthcare, followed by cultural heritage, finance, emergency management, autonomous systems, and academic knowledge organization. Typical applications included ontology enrichment, documentation, CQ generation, and schema extension. Biomedical ontologies such as SNOMED CT, UMLS, and the Gene Ontology were among the most frequently used datasets, while cultural and financial ontologies (e.g., DOREMUS, FIBO) also recurred across multiple studies.

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 various stages of the ontology development lifecycle, with research concentrated predominantly in the early and middle phases. Activities related to ontology implementation, particularly conceptualization and encoding, have received the greatest attention, together representing about 87.8% of the reviewed works, while later stages such as evaluation

and maintenance remain comparatively underexplored. In these core phases, LLMs demonstrated notable advantages by leveraging their strong natural language understanding and generative reasoning capabilities. They can automatically extract domain-specific concepts, infer hierarchical relations, and identify semantic patterns from unstructured text. Empirical evidence indicates that the resulting concept taxonomies often approximate expert-curated ontologies in terms of scalability and semantic coherence (Caufield et al. (2024); da Silva et al. (2024); Doumanas et al. (2024)), accelerating the creation of structured and high-quality knowledge representations in ontology development.

Despite notable advances, the application of LLMs across the ontology lifecycle remains uneven. Later-stage activities, such as documentation, evaluation, and maintenance, receive limited attention, as they demand capabilities that current LLMs cannot reliably provide. Evaluation requires strict logical consistency verification, which exceeds the intrinsic reasoning capacity of LLMs without external validation mechanisms such as rule checkers or expert review (Toro et al. (2024); Liu et al. (2025b)). Maintenance, in turn, depends on dynamic knowledge integration, whereas LLMs are statically trained and cannot incorporate new information without retraining, which limits their suitability for the long-term evolution of the ontology (Mundlamuri et al. (2025)). Furthermore, while early-stage tasks benefit from well-defined and quantifiable metrics, later stages often involve complex, less formalized objectives such as semantic coverage robustness and sustained ontology refinement.

Addressing these limitations requires reframing later-stage OE not as autonomous LLM-driven processes but as collaborative hybrid workflows. Future research should prioritize the development of hybrid architectures that integrate LLMs generated content with formal reasoning engines for constraint verification, the use of retrieval augmented generation techniques to maintain knowledge currency without full model retraining, and the design of human-centred workflows in which LLMs assist experts in validation and refinement rather than operating independently. Such approaches would leverage the generative flexibility of LLMs while preserving the analytical discipline and domain expertise essential for sustainable and trustworthy ontology engineering.

5.2 Configuration workflows of LLMs in ontology development activities

Our findings show that 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, aiming at reducing the manual effort and transfer of domain-specific expertise traditionally required by ontology engineers.

In the surveyed literature, a consistent trend can be observed regarding model selection and application. GPT-series models are predominantly employed for reasoning-intensive tasks, whereas open-source and lightweight models (e.g., LLaMA, Mistral) are increasingly favored for tasks like ontology matching and conceptualization. This reflects a rapidly diversifying LLM ecosystem where model choice is strategically aligned with task demands. For instance, tuned variants of LLMs like GPT-3 have been used to produce ontological constructs for knowledge formalization, while smaller models such as Mistral-7B offer faster inference and perform efficiently on smaller or domain-specific datasets.

Prompting techniques have a significant impact on performance in all models. Zero-shot prompting is widely used for its efficiency in well-defined tasks, while template-based prompting is essential for enforcing strict output schemas such as OWL axioms and SPARQL queries. Role-based prompting enhances semantic reliability in specialized domains, and more advanced strategies, including chain-of-thought reasoning, task decomposition, and iterative refinement, are adopted in heterogeneous or

multi-stage workflows to stabilize outputs and improve accuracy. Furthermore, the interaction between prompting and model behavior is further demonstrated by the input and output configurations of the OE workflows. LLMs can handle unstructured, semi-structured, and fully structured data, producing outputs ranging from natural language descriptions and competency questions to executable queries and formal axioms. Although natural language remains the most common input modality, there is a clear shift toward structured formats that better constrain model behavior and ensure the production of machine-actionable results. These structured formats often operate in tandem with the prompting strategies discussed above, constituting an integrated configuration approach that enhances the reliability and usability of LLM-generated ontological artifacts.

Building on these foundations, the integration of LLMs introduces more conversational and iterative workflows compared to traditional methodologies. LLMs enable broader participation from engineers, domain experts, and non-specialists through natural language inputs, which the models transform into ontology fragments, refinements, or validation feedback. This shift increases flexibility, accelerates development cycles, and improves the scalability and accessibility of OE practices.

Despite these advantages, several limitations remain. LLMs require substantial computational resources for access and fine-tuning, which restricts their scalability (Hoffmann et al. (2022); Treviso et al. (2023)). Their generalization across specialized domains is often poor unless guided by carefully designed prompts, and without such guidance they may produce incomplete or semantically irrelevant outputs (Barman et al. (2024)). Parameter adaptation methods, including full fine-tuning and parameter-efficient approaches such as Low-Rank Adaptation (LoRA), still demand considerable human expertise for data preparation, supervision, and quality control, thereby further increasing costs and resource limitations (Wang et al. (2025)). Compared with formal logic systems (Baader et al. (2017); Heindorf et al. (2022)), LLM reasoning abilities remain shallow, and issues such as hallucinations, limited transparency and violations of fundamental ontological constraints persist (Xu et al. (2025); Petroni et al. (2019); West et al. (2022); Huang et al. (2025)). These shortcomings require external validation, post-processing, and expert correction to ensure logical and semantic soundness.

Notably, only 5 studies in our review involve human participants in LLM-based OE tasks, revealing a clear gap in current research. Human experts remain essential because LLMs often struggle to accurately interpret specialized knowledge. Expert review and iterative validation are therefore necessary throughout OE tasks to ensure the accuracy, clarity, and overall reliability of the outputs. The limited use of human participation can be attributed to methodological and resource-related constraints, such as the high cost of involving expert participation, the difficulty in standardizing human-involved interaction workflows and the prevailing tendency to prioritize automation. Few studies that incorporate human input focus on tasks that require semantic judgment or complex reasoning, areas where LLMs remain less reliable. This pattern indicates that future research should integrate expert participation more systematically to maintain semantic and logical integrity and improve the reliability and usability of LLM-generated ontology output.

To address these limitations, a broader and more coordinated research agenda is required. Future work should emphasize hybrid neuro-symbolic architectures that integrate the generative capabilities of LLMs with the formal precision of symbolic reasoners, enabling continuous validation of logical constraints (Servantez et al. (2024); West et al. (2022); Hitzler et al. (2022)). Given the limited use of fine-tuning in current practice, an important direction is the development and adoption of parameter-efficient fine-tuning (PEFT) techniques (Wang et al. (2025)). Such approaches support more stable and focused adaptation of LLMs to ontological structures while avoiding the substantial cost of full model retraining.

In parallel, more robust prompting strategies that can adapt to evolving knowledge contexts are needed to mitigate hallucinations and semantic drift (Zhang et al. (2024b); Liu et al. (2025a)). To improve the scalability of validation processes, automated verification pipelines should combine ontology checks with streamlined expert oversight. To address the human involvement gap, future methodologies should establish clear and systematic frameworks for integrating human validation into LLM-supported workflows, reducing the cost of expert participation and incorporating human judgment into evaluation practices.

Finally, enhancing the transparency of LLMs remains an open challenge to build trust and support the long-term maintenance of the OE based on LLMs (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.

In general, achieving the full potential of LLMs in OE requires technical advances in both models and workflows, along with stronger human oversight, richer domain knowledge, and reliable formal verification.

5.3 Evaluation gaps and challenges for LLMs in ontology development activities

Our review shows that empirical validation has become a central practice in research on the use of large language models in ontology engineering. Nearly two-thirds of the surveyed task-level studies include full experimental evaluations, often built on open source domain ontologies in OWL or RDF that serve as expert-curated benchmarks. Most papers employ quantitative, qualitative or combined evaluation methods, reporting metrics such as precision, recall, F1 score and semantic similarity, while complementing these with expert assessments of conceptual soundness and domain relevance. It is important to note that these evaluations usually assess entire pipelines rather than isolating the contribution of the large language model component. Since each study adopts its own baselines and datasets, direct comparisons across papers are rarely meaningful. However, evaluation practices consistently indicate that the use of large language models increases automation and often improves task performance across several stages of the ontology engineering lifecycle.

Across the reviewed studies, we also observe a growing use of publicly available datasets, which supports the development of more reproducible evaluation frameworks. Shared ontologies increasingly function as common baselines that later research can replicate or extend, and several studies employ gold standard datasets to ensure fairness and comparability. Early efforts toward standardized and transparent evaluation protocols have begun to emerge. Initiatives such as the OAEI [11] and LLMs4OL (Giglou et al. (2024)) challenge explicitly define datasets, subtasks, and evaluation metrics, marking a move toward greater consistency and reproducibility within the field. More recently, dedicated benchmarking frameworks have been proposed for competency questions (Alharbi et al. (2024a)) and LLM-generated ontologies (Plu et al. (2024)), contributing to the growing infrastructure for standardized evaluation. From a methodological perspective, quantitative evaluations provide scalable, reproducible and transparent measurements of system performance (Liu (2011); Ioannidis and Maniadis (2024)). Qualitative assessments by domain experts capture semantic coherence, contextual relevance, and conceptual correctness that numerical metrics often overlook (Denzin et al. (2006); Patton (2014); Parfenova et al. (2025)). Integrating both

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

forms of evaluation combines statistical rigor with semantic depth, thereby helping to ensure that the resulting ontologies are not only formally sound but also contextually meaningful and usable.

Despite these advances and initial benchmarking efforts, several limitations remain. Existing initiatives are still fragmented and often target specific OE sub-tasks. Most studies define their own tasks, datasets, metrics, and benchmarks. This lack of uniformity makes the results difficult to compare, and even minor differences in prompt design or corpus selection can lead to result bias. More fundamentally, a coherent evaluation framework has yet to be established in the field. In addition, many studies do not standardize task definitions or input and output formats, which further complicates comparisons between different works. The lack of benchmark datasets for evaluating different ontology engineering tasks and the absence of clear and comprehensive evaluation metrics continue to constrain the development of the LLM-based OE community.

Another important limitation is that the performance of LLMs is often conflated with the behavior of the entire pipeline. Many studies assess only the final output, making it difficult to identify the model's actual strengths and weaknesses. Although the use of both quantitative metrics and expert evaluations has improved current practice, challenges remain. Quantitative metrics do not capture deeper semantic or domain-specific nuances, and qualitative assessments are time-consuming (Queirós et al. (2017)), require expertise and introduce subjectivity, which limits scalability.

To address these limitations, future research should prioritize the development of standard evaluation protocols for LLM-based OE. A first step is the creation of unified benchmarks with clearly defined datasets, tasks and metrics that enable consistent comparisons across studies and across different OE activities. Standardizing task definitions and the formats of inputs and outputs would further reduce variability and support greater reproducibility. In addition, modular evaluation frameworks (Wu and Yu (2024)) are needed to separate the contribution of the large language model from other components of the pipeline. Such frameworks would help evaluate specific capabilities, identify failure cases and provide a clearer understanding of model behavior. Evaluation metrics should also be refined to capture semantic correctness, conceptual validity and domain relevance, rather than relying mainly on surface level accuracy measures. More systematic error analysis would help to identify and address model issues.

Finally, new evaluation strategies should be explored to improve both depth and scalability. These may include automated semantic validation tools, structured expert review procedures, and hybrid approaches that combine statistical measures with targeted human validation. Together, these efforts can contribute to a more robust and reliable evaluation ecosystem for LLM-based OE.

5.4 Application domains of LLM-based ontology development

Across the reviewed studies, a clear trend emerges: while early applications were concentrated in healthcare and life sciences, the adoption of LLMs is rapidly expanding into domains such as cultural heritage, finance, emergency management, autonomous systems, and academic research. This highlights the inherent adaptability of models to address core ontology tasks from extraction and enrichment to validation and conceptual modeling in highly heterogeneous knowledge domains.

However, a cross-domain analysis reveals that the specific role of LLMs varies significantly from one domain to another. The uneven distribution of this progress provides important insight into the conditions that enable successful LLM–OE integration. For instance, Life sciences (Fathallah et al. (2024b)) and healthcare (Yang et al. (2023)) remain methodologically mature, supported by a powerful combination of factors such as abundant high-quality textual corpora (e.g., scientific literature, clinical documentation),

an urgent need for interoperability, and, critically, the availability of mature, gold-standard ontologies such as SNOMED CT and the Gene Ontology (GO). These well-established resources offer the structural scaffolding and authoritative examples needed to guide LLMs effectively, supporting tasks that include constraint validation and axiom generation.

In contrast, domains such as finance, disaster response, and cultural heritage often lack mature vocabularies and established development workflows. In these settings, LLMs are used less for refining existing ontologies and more for constructing domain knowledge from the beginning, including tasks such as knowledge extraction, conceptual modeling and ontology completion. Examples range from interpreting regulatory documents for financial ontologies (FIBO) to synthesizing search and rescue procedures in SAR ontology development. These studies show that LLMs can extract useful information from domain-specific resources and that expert validation helps improve the quality of the results. With access to large amounts of unstructured text, LLMs support domain experts in transforming natural language descriptions into initial conceptual structures.

LLMs have been consistently used as intermediaries that bridge unstructured text and formal representations, although their efficacy remains contingent upon the clarity of target schemas. Therefore, robust domain-specific adaptation remains a significant challenge (Mai et al. (2024)). Models trained on general corpora often struggle with specialized terminologies and evolving knowledge structures, leading to semantic imprecision. Furthermore, scalability issues arise because LLMs, being statically trained, struggle to dynamically incorporate new knowledge without retraining, which restricts their long-term applicability (Du et al. (2024)). Consequently, ensuring formal consistency in regulated domains still requires substantial expert validation (Perera and Liu (2024)).

To address these limitations, future research should focus on improving the ability of LLMs to adapt to specialized and evolving knowledge domains. This involves developing methods that support the creation and refinement of domain-specific vocabularies, schema templates and conceptual patterns, particularly in areas where consolidated ontologies are not yet available. At the same time, more effective mechanisms for integrating iterative expert feedback are needed so that domain specialists can actively shape and validate emerging conceptual structures throughout the development process. To ensure the long term applicability and accuracy of LLM-driven systems, techniques for dynamic knowledge updating and domain-aware adaptation are also essential. This includes continued advancement of continual learning strategies (Shi et al. (2025)) and dynamic update mechanisms (Fan et al. (2024)) that enable models to incorporate new terminology, regulatory changes and evolving domain understanding without requiring complete retraining.

By advancing these directions, the community can better leverage the generative scalability of LLMs while ensuring that the resulting ontological knowledge remains precise, reliable, and sustainable across domains with different levels of knowledge maturity.

6 Conclusion

Our study employs a systematic literature review methodology to examine the technical applications and current state of LLMs in OE. After searching multiple academic databases for literature published from 2018 to May 2025 using keywords related to LLMs and OE, 36 papers covering 49 independent task-level studies were selected through a multi-stage screening process. It should be noted that earlier Transformer-based studies not explicitly identified as language models may fall outside the scope of this review. Four

research questions (RQs 1–4) were formulated around the dimensions of LLM involvement in ontology development, focusing on supported core activities, technical implementation methods, performance evaluation strategies, and application domains. Key information was systematically extracted, including research context, details of LLM usage (roles, model types, prompting strategies, input/output formats, if human involved in OE tasks), evaluation settings, and target domains.

Across the 49 examined task-level studies, LLMs show clear strengths in early and middle stages of ontology development, especially in domain conceptualization, requirements specification, and ontology implementation. Models such as GPT and LLaMA, often used with zero-shot, template-based, or role-based prompts, can generate competency questions, formal axioms, and documentation. In these settings, they effectively take on responsibilities traditionally carried out by ontology engineers or domain experts. Their use in fields such as healthcare, cultural heritage, and autonomous systems illustrates the broad adaptability of current LLM-based approaches.

Although these findings highlight significant potential, several limitations remain. The support provided by current LLMs throughout the ontology lifecycle is uneven, and subsequent activities, such as documentation and long-term maintenance, receive comparatively little attention. Their reasoning remains shallow, often leading to hallucinated facts and limited transparency (Bakker et al. (2024); Manda (2025)), which requires expert correction to ensure logical and semantic soundness. Evaluation practices also present substantial difficulties. Existing studies rely on heterogeneous tasks, datasets, and metrics, leading to inconsistent and often incomparable results. Current evaluation measures capture only part of the semantic or conceptual quality of the generated content, and the lack of unified and contamination-free benchmark datasets restricts systematic comparison between studies (Paulheim (2025)).

These limitations are particularly pronounced in application domains that lack mature and stable ontological resources. In such contexts, vocabularies and schemas are still evolving, making it difficult for LLMs to interpret specialized terminology and preserve semantic consistency. Their static training further limits the timely incorporation of newly emerging knowledge, and in regulated or safety-critical settings, expert validation remains essential to ensure correctness.

Given these challenges, several research directions have become urgent:

- **Lifecycle Coverage Expansion**: Extend LLM applications to underrepresented ontology lifecycle stages, particularly documentation, maintenance, to ensure long-term sustainability and continuous evolution of ontology development.

- **Hybrid Neuro-Symbolic Reasoning**: Develop hybrid systems that integrate LLM-generated content with formal logic validation, including OWL reasoning, ontology constraint checking, and semantic consistency verification, improving semantic accuracy, maintaining constraint consistency, and reducing hallucinations.

- **Enhancing LLM Adaptability**: Improve prompt methods and reduce the reliance on structured inputs to make LLMs more adaptable in OE tasks. Using parameter-efficient fine-tuning (PEFT) can further help models adjust to ontological structures without the high cost of full retraining.

- **Standardized Evaluation Frameworks**: Establish reproducible benchmarks based on expert-curated and publicly documented datasets, and evaluation metrics that combine quantitative measures

with expert validation. Such expert-supported benchmarks are essential for reliably evaluating LLM-based OE systems, coping with dataset contamination and ensuring fair comparisons across different methods, ultimately contributing to a more robust and trustworthy evaluation ecosystem.

- **Continuous Learning and Dynamic Adaptation**: Develop domain-adaptive LLMs that can integrate evolving knowledge without requiring full retraining. This requires effective mechanisms for dynamic knowledge updates and domain-aware adaptation, supported by advances in continuous learning and dynamic update methods. These improvements help models maintain scalability and relevance in dynamic domains.

The dispersed nature of the reviewed tasks reflects the early stage of LLM-based OE research. As the field matures, we expect convergence toward more unified frameworks, shared resources, and standardized workflows.

By addressing these challenges, LLMs may progress from task-specific assistants to reliable collaborators in ontology engineering, supporting scalable, transparent, and high-quality knowledge representation across different domains. Achieving this vision will require not only technical innovation but also stronger methodological foundations, richer models of human and model interaction, and robust community standards.

7 Acknowledgments

This work was supported by the grant “SOEL: Supporting Ontology Engineering with Large Language Models” PID2023-152703NA-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF/UE”.

References

Achichi M, Lisena P, Todorov K, Troncy R and Delahousse J (2018) Doremus: A graph of linked musical works. Springer-Verlag. ISBN 978-3-030-00667-9. DOI:10.1007/978-3-030-00668-6_1.

Alharbi R, de Berardinis J, Grasso F, Payne T and Tamma V (2024a) Characteristics and desiderata for competency question benchmarks. In: *The Semantic Web-ISWC*.

Alharbi R, Tamma V, Grasso F and Payne T (2024b) 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 (2024c) Investigating open source llms to retrofit competency questions in ontology engineering. In: *Proceedings of the AAAI Symposium Series*, volume 4. pp. 188–198. DOI: 10.1609/aaaiss.v4i1.31793.

Alharbi R, Tamma V, Grasso F and Payne TR (2024d) The role of generative ai in competency question retrofitting. In: *European Semantic Web Conference*. Springer, pp. 3–13.

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. DOI:10.1007/978-3-031-19433-7_3.

Anisuzzaman D, Malins JG, Friedman PA and Attia ZI (2024) Fine-tuning llms for specialized use cases. *Mayo Clinic Proceedings: Digital Health* DOI:10.1016/j.mcpdig.2024.11.005.

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. DOI:10.1007/978-3-031-47745-4_16.

Arefeen MA, Debnath B and Chakradhar S (2024) Leancontext: Cost-efficient domain-specific question answering using llms. *Natural Language Processing Journal* 7: 100065. DOI:https://doi.org/10.1016/j.nlp.2024.100065. URL https://www.sciencedirect.com/science/article/pii/S294971912400013X.

Arevalo KMA, Ambre S and Dorsch R (2024) Autonto: Towards a semi-automated ontology engineering methodology. In: *International Knowledge Graph and Semantic Web Conference*. Springer, pp. 225–241.

Arslan M, Ghanem H, Munawar S and Cruz C (2024) A survey on rag with llms. *Procedia computer science* 246: 3781–3790.

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM and Sherlock G (2000) Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature genetics* 25(1): 25–9. DOI:10.1038/75556.

Azher IA, Seethi VDR, Akella AP and Alhoori H (2025) Limtopic: Llm-based topic modeling and text summarization for analyzing scientific articles limitations. In: *Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries*. New York, NY, USA: Association for Computing Machinery. ISBN 9798400710933. DOI:10.1145/3677389.3702605.

Baader F, Horrocks I, Lutz C and Sattler U (2017) *An introduction to description logic*. Cambridge University Press. DOI: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. DOI:10.1007/978-3-031-47240-4_22.

Bakker RM, Di Scala DL and de Boer M (2024) Ontology learning from text: an analysis on llm performance. In: *Proceedings of the 3rd NLP4KGC International Workshop on Natural Language Processing for Knowledge Graph Creation, colocated with Semantics*. pp. 17–19.

Balepur N, Gu F, Ravichander A, Feng S, Boyd-Graber JL and Rudinger R (2025) Reverse question answering: Can an LLM write a question so hard (or bad) that it can't answer? In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-190-2, pp. 44–64. DOI:10.18653/v1/2025.naacl-short.5. URL https://aclanthology.org/2025.naacl-short.5/.

Barman KG, Wood N and Pawlowski P (2024) Beyond transparency and explainability: on the need for adequate and contextualized user guidelines for llm use 26(3). DOI:10.1007/s10676-024-09778-2.

Bennett M (2013) The financial industry business ontology: Best practice for big data. *Journal of Banking Regulation* 14(3): 255–268. DOI:10.1057/jbr.2013.13.

Besta M, Blach N, Kubicek A, Gerstenberger R, Podstawski M, Gianinazzi L, Gajda J, Lehmann T, Niewiadomski H, Nyczyk P et al. (2024) Graph of thoughts: Solving elaborate problems with large language models. In: *Proceedings of the AAAI conference on artificial intelligence*, volume 38. pp. 17682–17690. DOI:10.1609/aaai.v38i16.29720.

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. DOI:10.1007/978-3-031-78952-6_13.

Bittner T, Donnelly M and Winter S (2005) Ontology and semantic interoperability. In: *Large-scale 3D data integration*. CRC Press, pp. 139–160. DOI:10.1007/978-3-031-39650-2_17.

Blomqvist E, Hammar K and Presutti V (2016) Engineering ontologies with patterns-the extreme design methodology. *Ontology Engineering with Ontology Design Patterns* 25: 23–50. DOI:10.3233/978-1-61499-676-7-23.

Bodenreider O (2004) The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research* 32(suppl_1): D267–D270. DOI:10.1093/nar/gkh061.

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 (2020) Language models are few-shot learners. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc. DOI:10.5555/3495724.3495883.

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. DOI: 10.1093/bioinformatics/btae104.

Chen Z, Zhou K, Zhang B, Gong Z, Zhao X and Wen JR (2023) ChatCoT: Tool-augmented chain-of-thought reasoning on chat-based large language models. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics. DOI:10.18653/v1/2023.findings-emnlp.985.

Ciatto G, Agiollo A, Magnini M and Omicini A (2025) Large language models as oracles for instantiating ontologies with domain-specific knowledge. *Know.-Based Syst.* 310(C). DOI:10.1016/j.knosys.2024.112940.

Cimmino A, Oravec V, Serena F, Kostelnik P, Poveda-Villalón M, Tryferidis A, García-Castro R, Vanya S, Tzovaras D and Grimm C (2019) Vicinity: Iot semantic interoperability based on the web of things. In: *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, pp. 241–247. DOI: 10.1109/DCOSS.2019.00061.

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. DOI:10.1016/j.websem.2024.100822.

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* : 100822DOI: 10.1016/j.websem.2024.100822.

Consortium GO (2006) The gene ontology (go) project in 2006. *Nucleic acids research* 34(suppl_1): D322–D326. DOI:10.1093/nar/gkj021.

Costa SD, Barcellos MP, de Almeida Falbo R, Conte T and de Oliveira KM (2022) A core ontology on the human–computer interaction phenomenon. *Data & Knowledge Engineering* 138: 101977. DOI:10.1109/ESEM56168.2023.10304795.

Coutinho ML (2024) Leveraging llms in text-based ontologydriven 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. DOI:10.1109/ETFA61755.2024.10710775.

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. DOI:10.1093/llc/fqq018.

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. DOI:10.1007/978-3-031-47243-5_17.

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.

Debnath T, Siddiky MNA, Rahman ME, Das P and Guha AK (2025) A comprehensive survey of prompt engineering techniques in large language models. *TechRxiv* DOI:10.36227/techrxiv.174140719.96375390/v2.

Denzin NK, Lincoln YS and Giardina MD (2006) Disciplining qualitative research. *International journal of qualitative studies in education* 19(6): 769–782.

Devlin J, Chang MW, Lee K and Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. pp. 4171–4186. DOI:10.18653/v1/N19-1423.

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. DOI:10.1186/s13326-016-0088-7.

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. DOI:10.1145/3688671.3688735.

Dong H, Chen J, He Y, Gao Y and Horrocks I (2024) A language model based framework for new concept placement in ontologies. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-60625-0, p. 79–99. DOI: 10.1007/978-3-031-60626-7_5.

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*. Association for Computing Machinery. ISBN 9798400701245, p. 5316–5320. DOI: 10.1145/3583780.3615126.

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. DOI:10.1007/978-3-031-63223-5_27.

Doumanas D, Soularidis A, Spiliotopoulos D, Vassilakis C and Kotis K (2025) Fine-tuning large language models for ontology engineering: A comparative analysis of gpt-4 and mistral. *Applied Sciences* 15(4): 2146. DOI: 10.3390/app15042146.

Drobnjakovic M, Kulvatunyou B, Ameri F, Will C, Smith B and Jones A (2022) The industrial ontologies foundry (iof) core ontology URL https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935068.

Du M, Luu AT, Ji B and Ng SK (2024) From static to dynamic: knowledge metabolism for large language models. AAAI Press. DOI:10.1609/aaai.v38i21.30564.

Duque-Ramos A, Fernández-Breis JT, Stevens R and Aussenac-Gilles N (2011) Oquare: A square-based approach for evaluating the quality of ontologies. *Journal of research and practice in information technology* 43(2): 159–176.

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. DOI:10.1007/978-3-031-71170-1_6.

Fan L, Hua W, Li L, Ling H and Zhang Y (2024) NPHardEval: Dynamic benchmark on reasoning ability of large language models via complexity classes. In: *Proceedings of the 62nd Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers). Bangkok, Thailand: Association for Computational Linguistics. DOI:10.18653/v1/2024.acl-long.225. URL https://aclanthology.org/2024.acl-long.225/.

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. DOI:10.1007/978-3-031-78952-6_4.

Fathallah N, Staab S and Algergawy A (2024b) LLMs4Life: Large language models for ontology learning in life sciences. In: *Proceedings of the EKAW 2024 Workshops, Tutorials, Posters and Demos*, CEUR Workshop Proceedings. URL https://ceur-ws.org/ELMKE_2024_paper_3.pdf.

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. DOI:10.1007/978-3-030-21348-0_29.

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. In: *Proceedings of the Workshop on Ontology Learning and Population*, *CEUR Workshop Proceedings*, volume 3577. URL https://ceur-ws.org/Vol-3577/paper16.pdf.

Gajulamandyam DK, Veerla S, Emami Y, Lee K, Li Y, Mamillapalli JS and Shim S (2025) Domain specific finetuning of llms using peft techniques. In: *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*. pp. 00484–00490. DOI:10.1109/CCWC62904.2025.10903789.

Gangemi A and Presutti V (2009) Ontology design patterns. In: *Handbook on ontologies*. Springer, pp. 221–243. DOI:10.1007/978-3-540-92673-3_10.

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. DOI:10.52825/ocp.v4i.2480.

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. DOI:10.1038/s41540-024-00358-0.

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. DOI:10.1007/978-3-031-07969-6_5.

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. DOI:10.52825/ocp.v4i.2485.

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* .

Gruninger M (1995) Methodology for the design and evaluation of ontologies. In: *Proc. IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing*.

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. DOI: 10.3233/AO-150157.

He Y, Chen J, Dong H and Horrocks I (2023) Exploring large language models for ontology alignment. In: *Proceedings of the ISWC 2023 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 22nd International Semantic Web Conference (ISWC 2023)*. CEUR Workshop Proceedings.

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. DOI:10.1145/3485447.3511925.

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. DOI:10.1145/3587259.3627571.

Hitzler P, Eberhart A, Ebrahimi M, Sarker MK and Zhou L (2022) Neuro-symbolic approaches in artificial intelligence. *National Science Review* 9(6): nwac035. DOI:10.1093/nsr/nwac035.

Hochreiter S and Schmidhuber J (1997) Long short-term memory. *Neural Computation* 9(8): 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Hoffmann J, Borgeaud S, Mensch A, Buchatskaya E, Cai T, Rutherford E, de Las Casas D, Hendricks LA, Welbl J, Clark A, Hennigan T, Noland E, Millican K, van den Driessche G, Damoc B, Guy A, Osindero S, Simonyan K, Elsen E, Vinyals O, Rae JW and Sifre L (2022) Training compute-optimal large language models. In: *Advances in Neural Information Processing Systems*. DOI:10.5555/3600270.3602446.

Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W et al. (2022) Lora: Low-rank adaptation of large language models. *ICLR* 1(2): 3.

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. DOI:10.1145/3703155.

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. DOI:10.1371/journal.pcbi.1000204.

Ioannidis JP and Maniadis Z (2024) Quantitative research assessment: using metrics against gamed metrics. *Internal and Emergency Medicine* 19(1): 39–47. DOI:10.1007/s11739-023-03447-w.

Janowicz K, Haller A, Cox SJ, Le Phuoc D and Lefrançois M (2019) Sosa: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics* 56: 1–10. DOI:https://doi.org/10.1016/j.websem.2018.06.003. URL https://www.sciencedirect.com/science/article/pii/S1570826818300295.

Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, de las Casas D, Bressand F, Lengyel G, Lample G, Saulnier L, Renard Lavaud L, Lachaux MA, Stock P, Le Scao T, Lavril T, Wang T, Lacroix T and El Sayed W (2023) Mistral 7b. *arXiv preprint arXiv:2310.06825* .

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 Trans. Softw. Eng. Methodol.* 33(7). DOI:10.1145/3672456.

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* : 15705838241304103DOI:10.1177/15705838241304103.

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

Kadam S and Vaidya V (2018) Review and analysis of zero, one and few shot learning approaches. In: *International Conference on Intelligent Systems Design and Applications*. Springer, pp. 100–112. DOI: 10.1007/978-3-030-16657-1_10.

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* .

Karras O (2024) Kg-empire: a community-maintainable knowledge graph for a sustainable literature review on the state and evolution of empirical research in requirements engineering. In: *2024 IEEE 32nd International Requirements Engineering Conference (RE)*. IEEE, pp. 500–501. DOI:10.1109/ESEM56168.2023.10304795.

Keet CM (2019) The african wildlife ontology tutorial ontologies. *Journal of Biomedical Semantics* 11. DOI: 10.1186/s13326-020-00224-y.

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 (2024) Sentimentgpt: Exploiting gpt for advanced sentiment analysis and its departure from current machine learning. In: *2024 IEEE International Conference on Big Data (BigData)*. pp. 7051–7060. DOI:10.1109/BigData62323.2024.10825350.

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. DOI:10.1016/j.infsof.2008.09.009.

Kojima T, Gu SS, Reid M, Matsuo Y and Iwasawa Y (2022) Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35: 22199–22213.

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. DOI:10.1017/S0269888920000065.

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. DOI:10.1007/978-3-319-46523-4_23.

Le H, Pino J, Wang C, Gu J, Schwab D and Besacier L (2021) Lightweight adapter tuning for multilingual speech translation. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 817–824. DOI:10.18653/v1/2021.acl-short.103.

Li J, Garijo D and Poveda Villalón M (2025) oeg-upm/llm4oe-slr: Release v1.0.0 — initial release of llm4oe-slr dataset and materials. DOI:10.5281/zenodo.15313672. URL https://doi.org/10.5281/zenodo.15313672.

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. DOI: 10.1007/978-3-031-78952-6_38.

Lippolis AS, Saeedizade MJ, Keskisärkkä R, Zuppiroli S, Ceriani M, Gangemi A, Blomqvist E and Nuzzolese AG (2025) Ontology generation using large language models. Springer-Verlag. ISBN 978-3-031-94574-8. DOI:10.1007/978-3-031-94575-5_18.

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: *The Semantic Web: 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 – June 2, 2022, Proceedings*. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-06980-2. DOI:10.1007/978-3-031-06981-9_23.

Liu H, Tam D, Muqeeth M, Mohta J, Huang T, Bansal M and Raffel CA (2022) Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems* 35: 1950–1965. DOI:10.5555/3600270.3600412.

Liu HH (2011) *Software performance and scalability: a quantitative approach*. John Wiley & Sons.

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. DOI: 10.1007/s40747-025-01833-9.

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*. Association for Computing Machinery. ISBN 9798400712746. DOI:10.1145/3696410.3714816.

Lo A, Jiang AQ, Li W and Jamnik M (2024) End-to-end ontology learning with large language models. In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc. ISBN 9798331314385. DOI:10.5555/3737916.3740683.

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. DOI:10.1007/978-3-031-77844-5_7.

Manda P (2025) Large language models in bio-ontology research: A review. *Bioengineering* 12(11): 1260. DOI: 10.3390/bioengineering12111260.

Marvin G, Hellen N, Jjingo D and Nakatumba-Nabende J (2023) Prompt engineering in large language models. In: *International conference on data intelligence and cognitive informatics*. Springer, pp. 387–402. DOI: 10.1007/978-981-99-7962-2_30.

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. DOI:10.1109/SYNASC61333.2023.00038.

Mienye ID, Swart TG and Obaido G (2024) Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information* 15(9). DOI:10.3390/info15090517. URL https://www.mdpi.com/2078-2489/15/9/517.

Miller GA (1995) Wordnet: a lexical database for english. *Communications of the ACM* 38(11): 39–41. DOI: 10.1145/219717.219748.

Mishra A, Shukla S, Torres J, Gwizdka J and Roychowdhury S (2025) Thought2Text: Text generation from EEG signal using large language models (LLMs). In: *Findings of the Association for Computational Linguistics: NAACL 2025*. Albuquerque, New Mexico: Association for Computational Linguistics. DOI:10.18653/v1/2025.findings-naacl.207. URL https://aclanthology.org/2025.findings-naacl.207/.

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.

Mundlamuri R, Gunnam GR, Mysari NK and Pujuri J (2025) The evolution of ai: From classical machine learning to modern large language models. *IEEE Access* 13: 178302–178341. DOI:10.1109/ACCESS.2025.3621344.

Norouzi SS, Mahdavinejad MS and Hitzler P (2023) Conversational ontology alignment with chatgpt. In: *Proceedings of the 18th International Workshop on Ontology Matching co-located with the 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 7, 2023*, *CEUR Workshop Proceedings*, volume 3591. CEUR-WS.org, pp. 61–66. URL https://ceur-ws.org/Vol-3591/om2023_STpaper1.pdf.

Noy NF and McGuinness DL (2001) Ontology development 101: A guide to creating your first ontology. *Knowledge Systems Laboratory* 32.

Olea C, Tucker H, Phelan J, Pattison C, Zhang S, Lieb M, Schmidt D and White J (2024) Evaluating persona prompting for question answering tasks. In: *Proceedings of th e 10th international conference on artificial intelligence and soft computing, Sydney, Australia*. DOI:10.5121/csit.2024.141106.

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.

Pan X, Ossenbruggen Jv, de Boer V and Huang Z (2024) A rag approach for generating competency questions in ontology engineering. In: *Research Conference on Metadata and Semantics Research*. Springer, pp. 70–81.

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. DOI:10.1109/ICDMW.2008.62.

Parfenova A, Marfurt A, Pfeffer J and Denzler A (2025) Text annotation via inductive coding: Comparing human experts to llms in qualitative data analysis. In: *Findings of the Association for Computational Linguistics: NAACL 2025*. pp. 6456–6469. DOI:10.18653/v1/2025.findings-naacl.361.

Parkkila J, Radulovic F, Garijo D, Poveda-Villalón M, Ikonen J, Porras J and Gómez-Pérez A (2017) An ontology for videogame interoperability. *Multimedia tools and applications* 76(4): 4981–5000. DOI: 10.1007/s11042-016-3552-6.

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.

Patton MQ (2014) *Qualitative research & evaluation methods: Integrating theory and practice*. Sage publications.

Paulheim H (2025) Towards evaluating knowledge graph construction and ontology learning with llms without test data leakage. In: *3rd workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM)*.

Pawlik M and Augsten N (2016) Tree edit distance: Robust and memory-efficient. *Information Systems* 56: 157–173.

Perera O and Liu J (2024) Exploring large language models for ontology learning. *Issues in Information Systems* 25: 299–310. DOI:10.48009/4_iis_2024_124.

Petroni F, Rocktäschel T, Riedel S, Lewis P, Bakhtin A, Wu Y and Miller A (2019) Language models as knowledge bases? In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics. DOI:10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250/.

Pinto HS, Staab S and Tempich C (2004) Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. pp. 393–397.

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.

Plu J, Escobar OM, Trouillez E, Gapin A and Troncy R (2024) A comprehensive benchmark for evaluating llm-generated ontologies. In: *The Semantic Web-ISWC*.

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.

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: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* DOI:10.5281/zenodo.887089.

Radford A and Narasimhan K (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.

Raiaan MAK, Mukta MSH, Fatema K, Fahad NM, Sakib S, Mim MMJ, Ahmad J, Ali ME and Azam S (2024) A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access* 12: 26839–26874. DOI:10.1109/ACCESS.2024.3365742.

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*. DOI:10.1145/3549737.3549765.

Rodríguez-Doncel V, Myles S, Iannella R and Steidl M (2018) ODRL vocabulary & expression 2.2. W3C recommendation, W3C. Https://www.w3.org/TR/2018/REC-odrl-vocab-20180215/.

Rumelhart DE, Hinton GE and Williams RJ (1986) Learning representations by back-propagating errors. *nature* 323(6088): 533–536. DOI:10.1038/323533a0.

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. DOI:10.1016/j.procs.2024.09.364.

Salamon JS and Barcellos MP (2022) Towards a framework for continuous ontology engineering. In: *ONTOBRAS*. pp. 158–165. DOI:10.1145/3422392.3422469.

Servantez S, Barrow J, Hammond K and Jain R (2024) Chain of logic: Rule-based reasoning with large language models. In: *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics. DOI:10.18653/v1/2024.findings-acl.159.

Shi H, Xu Z, Wang H, Qin W, Wang W, Wang Y, Wang Z, Ebrahimi S and Wang H (2025) Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys* 58(5): 1–42. DOI:10.1145/3735633.

Shin T, Razeghi Y, Logan IV RL, Wallace E and Singh S (2020) AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics. DOI: 10.18653/v1/2020.emnlp-main.346.

Staab S, Studer R, Schnurr HP and Sure Y (2001) Knowledge processes and ontologies. *IEEE Intelligent Systems* 16(1): 26–34. DOI:10.1109/5254.912382.

Straková J, Fucíková E, Hajic J and Uresová Z (2023) Extending an event-type ontology: Adding verbs and classes using fine-tuned llms suggestions. In: *Proceedings of the 17th linguistic annotation workshop (LAW-XVII)*. pp. 85–95.

Studer R, Benjamins VR and Fensel D (1998) Knowledge engineering: Principles and methods. *Data & knowledge engineering* 25(1-2): 161–197. DOI:10.1016/S0169-023X(97)00056-6.

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. DOI:10.1007/978-3-642-05151-7_16.

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. DOI:10.1007/978-3-031-60635-9_2.

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. DOI: 10.1109/ITSC57777.2023.10422308.

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)*. pp. 1–5. DOI:10.1109/ICASSP49357.2023.10094574.

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. DOI:10.1186/s13326-024-00320-3.

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 J, Ji T, van Aken B, Cao Q, Ciosici M, Hassid M, Heafield K, Hooker S, Raffel C, Martins P, Martins A, Forde J, Milder P, Simpson E, Slonim N, Dodge J, Strubell E, Balasubramanian N, Derczynski L, Gurevych I and Schwartz R (2023) Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics* 11: 826–860. DOI:10.1162/tacl_a_00577.

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. DOI: 10.1007/978-3-031-78952-6_9.

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. DOI:10.18653/v1/2024.climatenlp-1.13.

Vaithilingam P, Zhang T and Glassman EL (2022) Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. New York, NY, USA: Association for Computing Machinery. ISBN 9781450391566. DOI:10.1145/3491101.3519665.

Val-Calvo M, Aranguren ME, Mulero-Hernández J, Almagro-Hernández G, Deshmukh P, Bernabé-Díaz JA, Espinoza-Arias P, Sánchez-Fernández JL, Mueller J and Fernández-Breis JT (2025) Ontogenix: Leveraging large language models for enhanced ontology engineering from datasets. *Information Processing & Management* 62(3): 104042.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł and Polosukhin I (2017) Attention is all you need : 6000–6010DOI:10.5555/3295222.3295349.

Vieira ES and Gomes JA (2009) A comparison of scopus and web of science for a typical university. *Scientometrics* 81: 587–600. DOI:10.1007/s11192-009-2178-0.

Volz R, Kleb J and Mueller W (2007) Towards ontology-based disambiguation of geographical identifiers. In: *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.

Wang L, Chen S, Jiang L, Pan S, Cai R, Yang S and Yang F (2025) Parameter-efficient fine-tuning in large language models: a survey of methodologies. *Artificial Intelligence Review* 58(8): 227. DOI:10.1007/s10462-025-11236-4.

Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, Chi EH, Le QV and Zhou D (2022) Chain-of-thought prompting elicits reasoning in large language models. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713871088. DOI:10.5555/3600270.3602070.

West P, Bhagavatula C, Hessel J, Hwang J, Jiang L, Le Bras R, Lu X, Welleck S and Choi Y (2022) Symbolic knowledge distillation: from general language models to commonsense models. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics. DOI:10.18653/v1/2022.naacl-main.341. URL https://aclanthology.org/2022.naacl-main.341/.

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. DOI:10.1109/IEIR62538.2024.10959909.

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. DOI:10.1109/BigData59044.2023.10386743.

Wu Y (2024) *Large Language Model and Text Generation*. ISBN 978-3-031-55864-1, pp. 265–297. DOI:10.1007/978-3-031-55865-8_10.

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* 37(4): 1620–1634. DOI:10.1109/TKDE.2025.3536008.

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. DOI:10.1002/hcs2.61.

Yao S, Yu D, Zhao J, Shafran I, Griffiths T, Cao Y and Narasimhan K (2023) Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* 36: 11809–11822. DOI:10.5555/3666122.3666639.

Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan KR and Cao Y (2022) React: Synergizing reasoning and acting in language models. In: *The eleventh international conference on learning representations*.

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: *The Semantic Web: ESWC 2024 Satellite Events*. pp. 102–121. DOI:10.1007/978-3-031-78952-6_10.

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. In: *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics. DOI:10.18653/v1/2024.findings-acl.738. URL https://aclanthology.org/2024.findings-acl.738/.

Zhang S, Dong L, Li X, Zhang S, Sun X, Wang S, Li J, Hu R, Zhang T, Wang G et al. (2023a) Instruction tuning for large language models: A survey. *ACM Computing Surveys* DOI:/10.1145/3777411.

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 AJ (2023b) Multimodal chain-of-thought reasoning in language models. *Trans. Mach. Learn. Res.* .

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. DOI:10.1145/3639372.

Zheng M, Pei J, Logeswaran L, Lee M and Jurgens D (2024) When" a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. DOI:10.18653/v1/2024.findings-emnlp.888.

8 Appendix

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

8.1 LLM-supported Ontology Development Activities

Table 2 provides a comprehensive mapping of how LLMs contribute to specific ontology engineering activities across the 49 reviewed task-level studies. Each row represents one distinct study, 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	Prompt	Inputs	Outputs	Human involved
Requirements Specification – Functional Requirements Writing						
Fathallah et al. (2024a)	Ontology Engineer	GPT-3.5 LLaMA PaLM	Few-shot Role-based CoT	Natural language text	Natural language text	NO
Antia and Keet (2023)	Ontology Engineer	T5	N/A	Natural language text	CQs	NO
Requirements Specification – CQ Reverse Engineering						
Rebboud et al. (2024a)	Domain Experts	Not mentioned	N/A	Ontologies	CQs	NO
Alharbi et al. (2024b)	Ontology Engineer	GPT-3.5-turbo GPT-4 LLaMA2	Zero-shot	Triples	CQs	YES
Ciroku et al. (2024a)	Ontology Engineer	MiniLM T5 SBERT	Zero-shot Iterative refinement	KGs	CQs	NO

Table 2

Resource	Role	Model	Prompt	Inputs	Outputs	Human involved
Rebboud et al. (2024b)	Ontology Engineer	DPO [12] SOLAR[13] UNA [14] Zephyrβ GPT-3.5 GPT-4	Zero-shot Few-shot Fine-tuned FusionNet_7Bx2_MoE_14B Fine-tuned SOLAR-10.7B-Instruct-v1.0 Fine-tuned Mistral-7B Fine-tuned Mistral-7B-v0.1	Ontologies	CQs	NO
Alharbi et al. (2024c)	Ontology Engineer	GPT-3.5-turbo GPT-4 LLaMA-2-70B Mistral 7B Flan-T5-XL	Zero-shot Role-based Template-based	Triples	CQs	NO
Pan et al. (2024)	Domain Expert	GPT-4	Zero-shot Role-based	Natural language text	CQs	NO
Alharbi et al. (2024d)	Ontology Engineer	GPT-3.5-turbo GPT-4	Zero-shot Role-based Template-based	Triples	CQs	NO
Requirements Specification – Requirement Formalization						
Rebboud et al. (2024a)	Ontology Engineer	Not mentioned	N/A	Ontologies and CQs	Queries	NO
Tufek et al. (2024)	Ontology Engineer	ChatGPT	Zero-shot Template-based	Natural language text or CQs	SPARQL Queries	NO
Kholmska et al. (2024)	Ontology Engineer	ChatGPT Bard	Zero-shot Template-based Role-based Iterative refinement	Concepts	SPARQL Queries	NO
Ontology Implementation – Conceptualization						
Rebboud et al. (2024a)	Domain Experts	Not mentioned	N/A	CQs	Ontologies	NO
Goyal et al. (2024)	Ontology Engineer	LLaMA3 GPT-4o Mistral	Zero-shot Few-shot	Natural language text	Binary decision	NO
Coutinho (2024)	Ontology Engineer	Not mentioned	Zero-shot Few-shot CoT	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	Zero-shot Template-based Role-based Iterative refinement	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	Zero-shot Template-based Fine-tuned PLM	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	Zero-shot	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

[12] Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B
[13] SOLAR-10B-OrcaDPO-Jawade
[14] UNA-TheBeagle-7b-v1

Table 2

Resource	Role	Model	Prompt	Inputs	Outputs	Human involved
Toro et al. (2024)	Ontology Engineer	GPT-4 GPT-3.5-turbo	Few-shot Template-based	Term	JSON	NO
Pisu et al. (2024)	Ontology Engineer	BERT	Fine-tuned SciBERT	Natural language text	Relationships	NO
Val-Calvo et al. (2025)	Ontology Engineer	GPT-4	Iterative refinement	JSON	Schema	YES
Arevalo et al. (2024)	Ontology Engineer	GPT-4-1106-preview	Iterative refinement	Natural language text	JSON	NO
Doumanas et al. (2025)	Ontology Engineer	GPT-4 Mistral 7B	Iterative refinement	Natural language text	JSONL	YES
Ontology Implementation – Encoding						
Doumanas et al. (2024)	Ontology Engineer	GPT-4 GPT-3.5 Bard LLaMA	Zero-shot Template-based Iterative refinement	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	Few-shot Role-based CoT	Natural language text	CQs, Triples and Ontologies	NO
Caufield et al. (2024)	Ontology Engineer	OntoGPT	Zero-shot	Natural language text	Ontologies	NO
Eells et al. (2024)	Ontology Engineer	GPT-4	Zero-shot Template-based	Natural language text	Natural language text	NO
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	Zero-shot CoT GoT	CQs	Ontologies	NO
Mateiu and Groza (2023)	Ontology Engineer	GPT-3 Davinci model	Zero-shot Few-shot Fine-tuned GPT-3	Natural language text	Axioms	NO
Tang et al. (2023)	Ontology Engineer	ChatGPT	Zero-shot Template-based Iterative refinement Decomposition	Natural language text	Ontologies, JSON, Triples	NO
da Silva et al. (2024)	Ontology Engineer	GPT-4, Turbo4, Claude3, Gemini Pro	Zero-shot Template-based Few-shot One-shot	Natural language text, Ontologies	Ontologies	NO
Lippolis et al. (2024)	Ontology Engineer	GPT-4 Turbo API	N/A	CQs	Ontologies	NO
Val-Calvo et al. (2025)	Ontology Engineer Domain Expert	GPT-4	N/A	JSON and Schema	Ontologies	NO
Doumanas et al. (2025)	Ontology Engineer	GPT-4 Mistral 7B	Iterative refinement One-shot Role-based Fine-tuned GPT-4 Fine-tuned Mistral 7B	Natural language text CQs	Ontologies	NO
Ontology Development – Ontology Matching and Reuse						

Table 2

Resource	Role	Model	Prompt	Inputs	Outputs	Human involved
Zamazal (2024)	Domain Experts	GPT-4o	Zero-shot	Natural language text and verbalized candidates	Binary decision	NO
Kholmska et al. (2024)	Ontology Engineer	ChatGPT Bard	Zero-shot Template-based Role-based Iterative refinement	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	Zero-shot Few-shot	Ontologies and Natural language text	Mapping	NO
He et al. (2023)	Ontology Engineer	Flan-T5-XXL GPT-3.5-turbo	Zero-shot Template-based	Natural language text	Binary decision	NO
Norouzi et al. (2023)	Ontology Engineer	ChatGPT	Zero-shot	Natural language text	Mapping	NO
Ontology Development – Ontology Evaluation						
Tsaneva et al. (2024)	Domain Experts, Human Evaluator	GPT-4	Few-shot Zero-shot	Natural language text	Axioms	NO
Kholmska et al. (2024)	Ontology Engineer	ChatGPT Bard	Zero-shot Template-based Role-based Iterative refinement	Step 5: Ontologies	Step 5: Natural language text	NO
Fathallah et al. (2024a)	Domain Expert, Ontology Engineer	GPT-3.5 LLaMA PaLM	Few-shot Role-based CoT	Natural language text	Ontologies and Axioms	NO
Zhang et al. (2025)	Ontology Engineer	GPT-3	One-shot Few-shot Iterative refinement	Ontologies and CQs	Binary decision	YES
Ontology Publication – Documentation						
Bischof et al. (2024)	Domain Experts	Mistral 7B	Zero-shot Template-based Role-based	Natural language text	Terms	NO
Rebboud et al. (2024a)	Domain Experts	Not mentioned	N/A	Ontologies	Documentation	NO
Kholmska et al. (2024)	Ontology Engineer	ChatGPT Bard	Zero-shot Template-based Role-based Iterative refinement	Step 9: Ontology Extensions, Natural language text	Step 9: Documentation	NO
Fathallah et al. (2024a)	Ontology Engineer	GPT-3.5 LLaMA PaLM	Few-shot Role-based CoT	Natural language text, Ontologies	Documentation	NO
Giri et al. (2024)	Domain Experts, Ontology Engineer	T5	Fine-tuned T5	Terms	Documentation	NO
Maintenance – Bug Issue						
Kholmska et al. (2024)	Domain Expert	ChatGPT Bard	Zero-shot Template-based Role-based Iterative refinement	Step 8: Natural language text	Step 8: Natural language text	YES

8.2 Experimental Setup and Evaluation Overview

Table 3 summarizes the experimental validation practices across all 49 reviewed task-level 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 offer 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				
Fathallah et al. (2024a)	YES, a test[41]	Wine	N/A	N/A
Antia and Keet (2023)	YES[15]	Covid19 articles[16]	Qualitative	Human comment
Requirements specification – CQ Reverse Engineering				
Rebboud et al. (2024a)	N/A	DOREMUS, Polifonia, DemCare, Odeuropa, NORIA-O, FIBO	Quantitative	Cosine Similarity
Ciroku et al. (2024a)	YES[17]	WDV[18]	Quantitative	BLEU score
Rebboud et al. (2024b)	YES[40]	DOREMUS, Polifonia, Dem@Care, Odeuropa, NORIA-O, FIBO	Quantitative	Cosine Similarity
Alharbi et al. (2024b)	YES [49]	VideoGame, Dem@Care, VICINITY Core, African Wildlife	Hybrid	Number of CQs, Precision, Recall, F1
Alharbi et al. (2024c)	YES	Video Game, VICINITY Core, Dem@Care, Solar System Ontology	Quantitative	Number of CQs, Precision, Recall, F1
Pan et al. (2024)	YES[19]	KG-EmpiRE paper[20], Human–Computer Interaction paper[21]	Quantitative	Precision
Alharbi et al. (2024d)	YES [49]	VideoGame, Dem@Care, VICINITY Core, African Wildlife	Quantitative	Precision, Recall, F1
Requirements specification – Requirement formalization				
Rebboud et al. (2024a)	N/A	DOREMUS, Polifonia, DemCare, Odeuropa, NORIA-O, FIBO	Quantitative	Tree Edit Distance
Tufek et al. (2024)	YES[22]	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)	N/A	DOREMUS, Polifonia, DemCare, Odeuropa, NORIA-O, FIBO	Quantitative	Precision, Recall, F1, Accuracy, Consistent Ontology
Goyal et al. (2024)	YES[23]	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[24]	MM-S14-Disease, MM-S14-CPP	Quantitative	InRank@k, InRecall@k
Babaei Giglou et al. (2023)	YES[25]	WordNet, GeoNames, UMLS, National Cancer Institute, MEDCIN, SNOMEDCT US, Schema.org	Quantitative	MAP@K, F1-score
Toro et al. (2024)	YES[26]	Cell Ontology, UBERON, GO, Human Phenotype Ontology, Mammalian Phenotype Ontology, MONDO, Environment Ontology, Food Ontology, Ontology of Biomedical Investigations, Ontology of Biological Attributes	Hybird	Accuracy, Recall, F1, Manual Assessment
Pisu et al. (2024)	YES[27]	Computer Science Ontology	Quantitative	Accuracy, Precision, Recall, F1
Val-Calvo et al. (2025)	YES[42]	Airlines Customer Satisfaction[43], Amazon Ratings[44], BigBasket Products[45], Brazilian E-commerce[46], Customer Complaint[47], E-commerce Transactions[48]	Hybrid	N/A
Arevalo et al. (2024)	YES[28]	Computer Science Ontology	Hybrid	Mean aggregate similarity, Cosine similarity, Number of concepts, relations and properties, Dataset Size
Doumanas et al. (2025)	YES[50]	Search and Rescue Ontology	N/A	N/A

Table 3 – continued

Paper resource	Experiment	Data source	Evaluation type	Evaluation metric
Ontology implementation – Encoding				
Doumanas et al. (2024)	YES[29]	Search and Rescue Ontology	Hybrid	Analysis of False Positives, Precision, Recall, F1-score
Fathallah et al. (2024a)	YES, a test[41]	Wine	N/A	N/A
Caufield et al. (2024)	YES[30]	GO, EMAPA, MONDO Disease Ontology	Quantitative	F1, Precision, Recall
Eells et al. (2024)	YES[31]	101 nouns from COCA	N/A	N/A
Saeedizade and Blomqvist (2024)	YES[32]	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
da Silva et al. (2024)	YES[33]	CaSk	Hybird	Mean Error Score
Lippolis et al. (2024)	YES[34]	African Wildlife	Hybrid	OntoMetric
Val-Calvo et al. (2025)	YES[42]	Airlines Customer Satisfaction[43], Amazon Ratings[44], BigBasket Products[45], Brazilian E-commerce[46], Customer Complaint[47], E-commerce Transactions[48]	Hybrid	OQuaRE metrics, Pitfalls from OOPS!, Human reviews for terms
Doumanas et al. (2025)	YES [50]	Search and Rescue Ontology	Hybrid	Logical consistency, Hierarchy formation, Domain knowledge alignment, Precison, Recall, F1
Ontology development – Ontology matching and reuse				
Zamazal (2024)	YES[35]	EDOAL, Manchester from OAEI	Hybrid	Precision, Relaxed Precision, Recall
Kholmska et al. (2024)	N/A	OntoDM	N/A	N/A
Hertling and Paulheim (2023)	YES[36]	Ontologies from OAEI	Quantitative	Precision, Recall, F1, Size, Time
He et al. (2023)	YES[37]	NCIT-DOID, SNOMED-FMA	Quantitative	Precision, Recall, F1, Hits, MRR, RR
Norouzi et al. (2023)	YES	Ontologies from OAEI	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[41]	Wine	N/A	N/A
Zhang et al. (2025)	YES[38]	Music Meta	Qualitative	Feedback Scores
Ontology publication – Documentation				
Bischof et al. (2024)	N/A	N/A	Qualitative	Expert reviews
Rebboud et al. (2024a)	N/A	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[41]	Wine	N/A	N/A
Giri et al. (2024)	YES[39]	GO	Hybird	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

8.3 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 key studies that utilized them in our review. This

offers insights into how LLM applications vary in 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)

8.4 Ontology Datasets Used Across Studies

Table 5 lists all experiment datasets utilized in the reviewed task-level 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.

[15] https://github.com/pymj/AgOCQs
[16] https://github.com/pymj/AgOCQs/tree/main/AgOCQs/inputText
[17] https://github.com/King-s-Knowledge-Graph-Lab/revont
[18] https://github.com/gabrielmaia7/WDV
[19] https://github.com/XueliPan/GenCQs
[20] https://10.1109/ESEM56168.2023.10304795
[21] https://doi.org/10.1016/j.datak.2021.101977
[22] https://github.com/Siemens-OKE/llm-query-pipeline
[23] https://drive.google.com/drive/folders/1vRynlNH6LouIvcI1ymHsm6DwYKSOUoAa
[24] https://github.com/KRR-Oxford/LM-ontology-concept-placement
[25] https://github.com/HamedBabaei/LLMs4OL
[26] https://github.com/monarch-initiative/dragon-ai-results
[27] https://github.com/aleessiap/LeveragingLMforGeneratingOntologies
[28] https://github.com/Kiaramarnitt/AutOnto
[29] https://github.com/dimitrisdoumanas19/New-Experiments-LLMs.git
[30] https://github.com/monarch-initiative/ontogpt
[31] https://github.com/kastle-lab/commonsense-micropatterns
[32] https://github.com/LiUSemWeb/LLMs4OntologyDev-ESWC2024
[33] https://github.com/CaSkade-Automation/llm-capability-generation
[34] https://github.com/dersuchendee/Ontogenia
[35] https://github.com/OndrejZamazal/ComplexOntologyMatching-SEMANTiCS2024
[36] https://figshare.com/articles/code/OLaLa_for_OAEI
[37] https://github.com/KRR-Oxford/LLMap-Prelim
[38] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat
[39] https://github.com/kiharalab/GO2Sum

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
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	Odeuropa 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

Table 5

Acronym Name	Full Name	URL	Domain
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
GeoNames	GeoNames Geographical Database	https://www.geonames.org/	Geography
WordNet	Princeton WordNet Lexical Database	https://wordnet.princeton.edu/homepage	Lexical Semantics
COCA	Corpus of Contemporary American English	https://www.english-corpora.org/coca/	Linguistics
COVID-19 Articles (AgOCQs)	COVID-19 Article Corpus for AgOCQs	https://github.com/pymj/AgOCQs/tree/main/AgOCQs/inputText	Healthcare Text Mining
KG-EmpiRE Corpus	KG-EmpiRE Paper Text Corpus	https://10.1109/ESEM56168.2023.10304795	Computer Science
HCI Corpus	Human–Computer Interaction Paper Corpus (used in GenCQs)	https://doi.org/10.1016/j.datak.2021.101977	Computer Science
Airline-CS	Airline Customer Satisfaction Dataset	https://www.kaggle.com/datasets/raminhuseyn/airline-customer-satisfaction	Transportation
Amazon Ratings	Amazon Ratings Dataset	https://labur.eus/voslX	E-commerce
BigBasket Products	BigBasket Products Dataset	https://labur.eus/tnqzt	E-commerce
Brazilian E-commerce	Brazilian E-commerce Dataset	https://labur.eus/WM3cS	E-commerce
Customer Complaint	Customer Complaint Dataset	https://labur.eus/NlmH7	Customer Service
E-commerce Transactions	E-commerce Transactions Dataset	https://labur.eus/AJ8bF	E-commerce

[40] https://github.com/D2KLab/llm4ke
[41] https://github.com/andreamust/NEON-GPT
[42] https://github.com/tecnomod-um/OntoGenix
[43] https://labur.eus/SN5n1
[44] https://labur.eus/voslX
[45] https://labur.eus/tnqzt
[46] https://labur.eus/WM3cS
[47] https://labur.eus/NlmH7
[48] https://labur.eus/AJ8bF
[49] https://github.com/SemTech23/RETROFIT-CQs
[50] https://github.com/dimitrisdoumanas19/Fine-tuning-LLMS