Ontology Construction and Reasoning with LLMs: Survey and Perspective

Hui Yang
The University of Manchester, UK

Jiaoyan Chen
The University of Manchester, UK

Yue Ma
Université Paris-Saclay, CNRS, LISN, France

Renate Schmidt
The University of Manchester, UK

Abstract

Ontologies provide precise semantics and reliable reasoning capabilities and are widely used in real-world domains such as healthcare and finance. However, traditional ontology construction and reasoning face several limitations: ontology construction typically requires substantial human effort, while standard reasoning approaches struggle with incomplete or uncertain information. To address these challenges, the use of large language models (LLMs) in ontology construction and reasoning has gained significant attention. Despite a growing body of research, the field lacks a systematic review, which hinders researchers from obtaining a comprehensive understanding of recent advances. To bridge this gap, we survey recent work, primarily since 2021, and categorize the current research landscape into three areas: Ontology Learning, Ontology Matching, and Ontology-aware Complex Reasoning. We examine how ontologies are integrated with LLMs for knowledge extraction, interaction, and reasoning through approaches such as prompting, fine-tuning, and retrieval-augmented generation (RAG). Finally, we discuss current challenges and outline promising directions for future research.

2012 ACM Subject Classification Information systems → Web Ontology Language (OWL); Computing methodologies → Description logics; Computing methodologies → Natural language generation

Keywords and phrases Description Logic, Ontology, LLM

Digital Object Identifier 10.4230/TGDK.1.1.42

Funding This work is funded by the EPSRC project OntoEm (EP/Y017706/1)

Received Date of submission **Accepted** Date of acceptance **Published** Date of publishing

Editor TGDK section area editor

1 Introduction

Ontologies [10], especially those expressed in the Web Ontology Language (OWL), offer a formal framework for representing domain knowledge through concepts, relations, and constraints. They enable knowledge sharing, automated reasoning, and explainable intelligent systems in domains such as biomedicine [7] and healthcare [29]. For instance, a medical ontology may define classes such as *Disease*, *Pneumonia*, and *Lung*, relations such as *locatedIn*, and axioms such as "pneumonia is a lung disease," formalised as $Pneumonia \sqsubseteq Disease \sqcap \exists locatedIn.Lung$. These representations support tasks such as consistency checking, inference, classification, and heterogeneous data integration. Although large language models (LLMs) have achieved substantial success, ontologies remain essential because they provide formal semantics, logical consistency, and machine-interpretable structure. They therefore complement, rather than replace, neural approaches.

Traditionally, ontology construction has been a knowledge-engineering task performed by domain experts and knowledge engineers [64]. Although this expert-driven process can produce

© Hui, Jiaoyan, Yue and Renate;
licensed under Creative Commons License CC-BY 4.0

high-quality and logically coherent ontologies, it is time-consuming, labour-intensive, and difficult to scale, particularly in large, complex, or rapidly evolving domains. To reduce this burden, ontology learning seeks to automatically or semi-automatically extract ontological components, including concepts, relations, and axioms, from text corpora [15, 8, 5]. A common line of work in ontology learning relies on linguistic patterns, statistical association measures, clustering, supervised classification, or rule-based and symbolic techniques to identify candidate concepts and relations from textual data. However, these approaches still face several limitations. First, their outputs often contain substantial noise and low-precision candidates, which may require further validation and curation. Second, supervised approaches require labelled training data and may have limited ability to generalise across different domains.

The rapid development of Transformer-based language models has opened new opportunities for ontology learning and construction. Early encoder-based models, such as BERT [26] and RoBERTa [52], demonstrated that large-scale pre-training on unlabelled text can produce powerful contextual representations that are adaptable to a wide range of downstream tasks. Subsequent decoder-based and sequence-to-sequence models, including GPT-2 [72], T5 [73], and GPT-3 [18], enabled training at larger scales and on increasingly diverse corpora, leading to substantial improvements in language generation. In particular, models such as GPT-3 showed emerging capabilities in few-shot prompting and in-context learning. More recent frontier and open-weight LLMs, including GPT-5.5 [66], Gemini 3.1 Pro [35], recent Claude models such as Claude Fable 5, Claude Mythos 5, and Claude Opus 4.8 [4], Qwen3.6 [70, 71], DeepSeek-V4 [25], and Llama 4 [57], have further advanced long-context processing, multimodal understanding, tool use, agentic workflows, and complex reasoning. These developments suggest that generative LLMs are increasingly capable of supporting knowledge-intensive tasks, including knowledge elicitation, relation discovery, ontology enrichment, and cross-domain knowledge application.

In this survey, we aim to provide a comprehensive analysis of recent studies on ontology construction and reasoning with large language models (LLMs). Specifically, we categorize existing work into three main areas:

1. **Ontology learning** (Section 2), which focuses on studies that use LLMs to construct or complete ontologies either by extracting knowledge from various sources or by directly generating ontological structures. This category includes taxonomy completion (Section 2.1) and taxonomy construction (Section 2.2), where a taxonomy can be regarded as a simplified ontology consisting only of concepts and their hierarchical relations. It also covers related tasks such as relationship learning (Section 2.3) and ontology population (Section 2.4).
2. **Ontology matching** (Section 3), also known as ontology alignment, which aims to identify correspondences between concepts across two given ontologies for the purposes of integration and reuse. These correspondences may involve specific semantic relations, such as equivalence or subsumption. In this part, we discuss different types of approaches that employ LLMs in various ways, for example as matchers, agents, components in end-to-end ontology matching pipelines, or tools for semantic enrichment (Section 3.1).
3. **Ontology-aware complex reasoning** (Section 4), which focuses on non-standard reasoning tasks. This includes LLM-based reasoning over natural language with the support of ontologies (Section 4.1), as well as the use of LLMs for ontology reasoning tasks such as deductive reasoning, abductive reasoning, and consistency checking (Section 4.2).

At the end of each part, we provide a detailed discussion of existing challenges and potential directions for future research.

To the best of our knowledge, this is the first survey that focuses specifically on methodological advances in ontology construction, ontology matching, and ontology-aware reasoning with LLMs. A

Table 1 Summary of papers discussed in Ontology Learning, ordered by publication year. Abbreviations: Taxo. = taxonomy; comp. = completion; gen. = generation; Onto. = ontology; pop. = population.

Paper	Task	Method	Models
GenTaxo [99] (Zeng et al. 2021)	Taxo. comp.	Prompting	PLM
TacoPrompt [89] (Xu et al. 2023)	Taxo. comp.	Prompting	BERT
[60] (Nikishina et al. 2023)	Taxo. comp.	Prompting + fine-tuning	PLM
LLMs4OL [32] (Giglou et al. 2023)	Taxo. gen.; relation learning	Prompting + fine-tuning	BERT T5 LLaMA
[30] (Funk et al. 2023)	Taxo. gen.	Prompting	GPT
[22] (Chen et al. 2023)	Taxo. gen.	Prompting + fine-tuning	GPT LLaMA
SPIRES [20] (Caufield et al. 2023)	Onto. pop.	Prompting	GPT
ICON [80] (Shi et al. 2024)	Taxo. comp.	Prompting	GPT
OLLM [54] (Lo et al. 2024)	Taxo. gen.	Fine-tuning	Mistral
CHIME [42] (Hsu et al. 2024)	Taxo. gen.	Prompting	T5
[82] (Sun et al. 2024)	Taxo. gen.	Prompting	GPT Claude LLaMA
Chain-of-Layer [97] (Zeng et al. 2024)	Taxo. gen.	Prompting	GPT
TaxoLLaMA [59] (Moskvoretskii et al. 2024)	Taxo. comp.	Fine-tuning	LLaMA
[28] (Dong et al. 2024)	Concept placement	Prompting + fine-tuning	BERT T5 GPT
[56] (Mai et al. 2024)	Taxo. gen.; relation learning	Fine-tuning	Falcon GPT LLaMA
[62] (Norouzi et al. 2024)	Onto. pop.	RAG	GPT LLaMA
CodeTaxo [98] (Zeng et al. 2025)	Taxo. comp.	Prompting	GPT
GeTT [87] (Wu et al. 2025)	Taxo. gen.	Prompting	SBERT
LITE [101] (Zhang et al. 2025)	Taxo. eval.	Prompting	GPT
[47] (Kollapally et al. 2025)	Taxo. comp.	Prompting	GPT
[9] (Ayad et al. 2025)	Onto. pop.	Prompting	GPT
OntoAxiom [11] (Bakker et al. 2025)	Onto. eval.	Prompting	Qwen LLaMA GPT
OntoTune [53] (Liu et al. 2025)	Taxo. comp.; QA	Fine-tuning	LLaMA
TaxoPro [90] (Xu et al. 2025)	Taxo. comp.	Fine-tuning	BERT
COMI [91] (Xu et al. 2025)	Taxo. comp.	Prompting + fine-tuning	LLaMA BERT
[75] (Sadowski et al. 2026)	Onto. pop.	Prompting	GPT Claude Gemini Qwen DeepSeek Kimi

Model icons: PLM; BERT BERT; SBERT SBERT; T5 T5/FLAN-T5; GPT/ChatGPT; LLaMA; Claude; Mistral; Qwen; Falcon/TII; Gemini; DeepSeek; Kimi; GLM GLM; Phi Phi; Aya Aya; InternLM InternLM; DLM Domain.

related survey [48] also investigates the use of LLMs with ontologies; however, its focus is primarily on the application of LLMs across the ontology engineering lifecycle, including requirements specification, implementation, publication, and maintenance. In contrast, our focus is different: we concentrate on methodological advances in automated ontology construction and reasoning, while also covering broader topics such as ontology matching and complex reasoning with LLMs.

2 Ontology Learning

Ontology learning aims to automatically or semi-automatically acquire ontological knowledge from diverse sources, including texts, tables, existing taxonomies, knowledge graphs, and the parametric knowledge encoded in language models.

This survey organizes LLM-based ontology construction work into four subtasks: taxonomy completion (Section 2.1), taxonomy construction (Section 2.2), relationship learning (Section 2.3), and ontology population (Section 2.4). These subtasks differ in both their input assumptions and the type of ontological knowledge they target. Taxonomy completion repairs or enriches an existing hierarchy of concepts, while taxonomy construction builds a new taxonomy directly using LLMs or from given sources such as documents and tables.[1] Relationship learning extends

[1] E.g., a taxonomy can be represented by hierarchical relations such as Dog $\sqsubseteq$ Animal, meaning that every dog is a type of animal.

beyond hierarchical structure to extract non-taxonomic relations and richer axiom types.[2] Ontology population, by contrast, aims to instantiate a given ontology schema by generating ABox assertions, namely the assertional component of an ontology that describes individuals, their class memberships, and their relationships.[3]

Across these subtasks, recent LLM-related studies commonly adopt three methodological paradigms, either individually or in combination (Section 2.5): (1) *prompting*, which guides LLMs through carefully designed input instructions; (2) *fine-tuning*, which adapts pretrained models to specific tasks or domains using additional training data; and (3) *retrieval-augmented generation (RAG)*, which incorporates external knowledge retrieved from relevant sources into the generation process.

Table 1 summarizes the surveyed studies according to the subtask addressed and the LLM-based method employed.

2.1 Taxonomy Completion

A taxonomy can be regarded as a simple ontology consisting only of atomic concepts and their hierarchical relations. It can also be modeled as a Directed Acyclic Graph (DAG), where each node represents a concept and each edge $A \rightarrow B$ denotes a hierarchical relation in which B is a subclass of A, formally represented as $B \sqsubseteq A$ in the syntax of Description Logics [10].

Existing taxonomy completion methods can be broadly categorised into two classes: (1) *concept subsumption completion*, which aims to identify missing parent or child concepts for a given concept; and (2) *concept placement*, which focuses on inserting new concepts into an existing taxonomy to complete or enrich its structure.

2.1.1 Concept Subsumption Completion

Concept subsumption completion aims to identify subsumption relations for given concepts. In previous studies, this problem has also been formulated as *hypernym/hyponym Discovery*, which focuses on identifying parent or child concepts of a given concept.

Nikishina et al. [60] is one of the early works that attempts to solve hypernym and hyponym discovery given a concept and relevant background information. The authors explore both in-context learning, such as zero-shot and few-shot learning, and the fine-tuning of LLMs. As a follow-up work, TaxoLLaMA [59] extends this line of research by constructing a new fine-tuning dataset based on WordNet, a general-domain taxonomy. It also broadens the task scope to include taxonomy construction and Lexical Entailments, i.e., discovering subsumption relations between phrase pairs.

OntoTune [53] further extends these ideas by proposing a domain-specific LLM fine-tuning method that enhances domain-specific capabilities while preserving general-domain abilities. The main idea is to use domain ontologies, such as SNOMED CT in the healthcare domain, to construct fine-tuning datasets. These datasets are obtained by collecting domain information provided by the ontology, including concept definitions, hypernyms, and synonyms. The resulting data is then used to fine-tune the model through methods such as supervised instruction fine-tuning (SFT), direct preference optimization (DPO), or combinations of both. This improves the model's performance on domain-specific tasks, including hypernym discovery and more complex tasks such

[2] For example, the axiom $\mathsf{Parent} \sqsubseteq \exists \mathsf{hasChild}.\mathsf{Person}$ states that every parent has a child who is a person.
[3] For example, an ABox assertion such as $\mathsf{City}(\mathsf{Paris})$ states that Paris is an instance of the class City; an assertion such as $\mathsf{locatedIn}(\mathsf{Paris}, \mathsf{France})$ states that Paris is located in France.

as question answering over the corresponding domain, without substantially compromising its general capabilities.

2.1.2 Concept Placement

Concept placement aims to insert a new concept into an appropriate position in an existing taxonomy. Early work, such as TaxoExpan [79], primarily addressed the hypernym-only setting, in which the new concept is added as a leaf node and only its parent needs to be identified. By contrast, this survey focuses on more recent formulations of the task, first introduced by TMN [100], where the concept may be inserted into an internal position of the taxonomy, requiring consideration of both its hypernym and hyponym relations.

We first discuss methods that assume the new concept is given in advance, and then review approaches that additionally aim to discover missing concepts.

Given New Concept. Formally, given a new concept A, the goal is to identify all existing taxonomy edges "$Parent \rightarrow Child$" such that replacing each edge with "$Parent \rightarrow A \rightarrow Child$" yields valid hierarchical relations.

Earlier studies on taxonomy enrichment mainly rely on neural representation learning. These methods learn embeddings for query concepts, candidate edges, and their structural or textual contexts, and then estimate the compatibility between a query concept and a candidate insertion position through task-specific scoring functions [100, 45, 6, 61]. More recently, geometric-based methods have been proposed to encode hierarchical structure more explicitly through geometric relations, especially box-based inclusion relations, which provide a natural way to model parent-child semantics in taxonomy space [46, 92, 58]. Since this survey focuses on LM-based methods, we only briefly position these non-LM approaches here and do not provide a detailed discussion.

In contrast to embedding-based neural methods above, LM-based methods employ language models more directly as scoring, prediction, or adaptation modules. For example, TacoPrompt [89] formulates edge scoring as a probability estimation problem with prompt learning based on pre-trained BERT, where the query concept and candidate edge are jointly encoded in the input context. COMI [91] follows the general vector-based paradigm, but uses prompt learning to directly encode textual information into fixed vector representations, which are then used for subsequent LLM fine-tuning on the taxonomy enrichment task. TaxoPro [90] focuses on the low-resource setting and transfers knowledge from high-resource taxonomies through a plug-in LoRA-based strategy, demonstrating the effectiveness of model adaptation when target-domain supervision is limited. CodeTaxo [98] represents concept taxonomies as code and leverages code-oriented language prompts to improve taxonomy completion. Similarly, Dong et al. [28] apply LLMs through either fine-tuning or prompt learning to select plausible candidate edges. Unlike most existing studies, however, their work considers complex $\mathcal{EL}$ ontology concepts and incorporates contextual information beyond concept names. They further construct a new benchmark for this setting using clinical records and the widely used healthcare ontology SNOMED CT.

Unknown New Concept. We consider a more general setting in which the target concept is not provided in advance and must instead be discovered by the model. Given a taxonomy, the model is required to first identify a missing concept and then insert it into the appropriate position within the taxonomy. Two main semantic-based strategies have been studied, as summarized below.

GenTaxo [99] proposes a three-stage framework for this task. First, for each taxonomy edge, it learns a fused embedding that captures both structural and semantic information from neighboring concepts using PLMs and GNNs. Second, a position classifier is trained on these fused embeddings to determine whether a valid missing concept exists between the connected nodes. Finally, when the classifier predicts the presence of a missing concept, a concept generator produces candidate

concepts using the fused embedding as input.

In contrast, ICON [80] follows the reverse order: it first identifies missing concepts and then inserts them into the taxonomy. Specifically, ICON detects potential missing concepts using clustering methods and employs prompt learning to generate candidate class names and definitions. The newly identified concepts are then inserted into their appropriate positions within the taxonomy.

2.2 Taxonomy Construction

Taxonomy construction aims to build a complete taxonomy either from a given set of concepts, with the support of external sources, or directly through LLM generation. Based on the source of taxonomic information, we categorize existing approaches into three types as follows.

Type 1: Constructing a taxonomy over given concepts. Chain-of-Layer [97] proposes an iterative method that uses LLMs to construct a taxonomy from a given set of concepts in a layer-by-layer manner. Specifically, the method prompts an LLM to generate parent–child relations and then applies an ensemble-based ranking filter, built on SciBERT, to filter and repair the generated edges. The method can also operate in a zero-shot setting by using an LLM-generated taxonomy as an in-context example.

Chen et al. [22] provide a systematic comparison of taxonomy construction methods based on prompting and fine-tuning. Both types of methods follow the same preprocessing pipeline to generate the final taxonomy. The results show that prompting-based approaches outperform fine-tuning-based approaches, and that the performance gap becomes larger when the training dataset is small, with F1-score improvements ranging from 2% to 20%. However, fine-tuning-based approaches can be more easily post-processed to satisfy structural constraints, such as requiring each node to have exactly one parent. This remains more challenging for prompting-based methods, where 1.75%–4.39% of nodes in the WordNet dataset fail to satisfy this constraint.

Bakker et al. [11] present a systematic evaluation of LLMs for ontology learning, including the prediction of subclass axioms, which tests whether LLMs can recover a taxonomy from a given set of concepts. Based on evaluations over nine medium-sized ontologies, the study finds that LLMs achieve satisfactory performance on general-domain ontologies, such as FOAF, with F1-scores of up to 0.642. However, performance drops substantially in more specialized domains. For example, on an ontology focused on railways, the F1-score decreases to approximately 0.055.

Type 2: Taxonomy extraction or learning from external resources. OLLM [54] proposes constructing ontologies from documents by fine-tuning LLMs to generate taxonomy paths conditioned on input documents. The generated paths from different documents are subsequently merged and pruned using heuristic rules to obtain the final ontology. Notably, OLLM assumes that multiple documents containing relevant information are provided as input, and it aggregates information across these documents into a unified ontology. Unlike Funk et al. [30], whose evaluation relies largely on human experts, OLLM evaluates the generated taxonomies using automatic metrics, including fuzzy, continuous, and graph-based F1 scores, as well as Motif Distance, thereby better capturing both structural and semantic quality. It also demonstrates transfer learning ability by training the model on Wikipedia and evaluating it on the ArXiv dataset. Other studies, such as Zhang et al. [101], have proposed alternative metrics for more sophisticated analysis of taxonomy quality, aiming to make evaluation more efficient, faithful, and consistent.

LLMs4OL [32] provides a systematic evaluation of LLMs' capabilities for ontology learning from three perspectives: (1) term typing, (2) taxonomy discovery, and (3) non-taxonomic relation extraction. However, its evaluation mainly focuses on the ability of LLMs to determine whether given pairs satisfy specific ontology-related relations across these three tasks. For example, in

the taxonomy discovery task, positive and negative pairs (A, B) are constructed such that A is, or is not, a subclass of B, and standard classification metrics such as F1-score are used to evaluate performance. The results show that models such as GPT-4 and GPT-3.5 exhibit reasonable taxonomy discovery capabilities, with F1-scores ranging from 67.8 to 78.1. Moreover, after fine-tuning, smaller open-source models can outperform larger proprietary models in some cases; for instance, on the schema.org dataset, a fine-tuned model achieves an F1-score of up to 91.7, compared with 74.4 for GPT-3.5. Mai et al. [56] conduct a similar evaluation, but focus on different domains. They report a similar observation: in specific domains, the performance of LLMs on ontology learning drops significantly, but fine-tuning can substantially improve performance. For example, on the gibberish WN-Sweets dataset, Falcon-7B improves from an F1-score of 0.352 to 0.655 after fine-tuning.

In addition, several other works aim to construct taxonomies from different data sources or for specific domains. For example, GeTT [87] constructs taxonomies by prompting LLMs to generate paths, followed by path merging and pruning. However, instead of using documents, GeTT learns taxonomies from table repositories. Specifically, given a set of tables, it first identifies the corresponding concept of each table and then constructs a taxonomy over these concepts. CHIME [42] investigates the task of generating taxonomies of scientific studies for literature review reports. Given a collection of studies on a research topic, CHIME first generates top-level root categories and then constructs a taxonomy for each root category, both through LLM prompting. Finally, Kollapally et al. [47] investigate taxonomy generation in the Social Determinants of Health domain by extending an existing taxonomy to SOHOv2 through the extraction of fact triples from PubMed articles.

Type 3: Direct generation by LLMs. Funk et al. [30] investigate the task of directly generating a taxonomy using LLMs, with only a given seed root concept as input. Their method follows an iterative procedure: for each selected concept, the LLM first determines whether subconcepts exist, then generates candidate subconcepts and inserts them into the taxonomy. However, their evaluation relies primarily on human experts, and the overall evaluation scale remains relatively limited. Sun et al. [82] study a related question from an application perspective by evaluating whether LLMs can replace curated taxonomies. Their analysis is useful for ontology learning because it suggests that LLMs can provide flexible semantic signals, but they do not yet provide the stable, explicit, and reusable structure offered by taxonomies.

2.3 Relationship Learning

Unlike taxonomy construction or completion, which primarily focuses on hierarchical "is-a" relations, relationship learning aims to discover broader semantic relations between concepts. A relation can be represented as a triple such as $(Drug, treats, Disease)$, where relation $r = treats$ indicates that a drug may treat a disease. In description logic, such a relation can also be expressed as axioms of the form $A \sqsubseteq \exists r.B$. For example, $Course \sqsubseteq \exists taughtBy.Instructor$ with relation $r = taughtBy$ means that every course is taught by at least one instructor. Therefore, relationship learning goes beyond "is-a" relations and captures richer semantic connections between concepts.

LLMs4OL [32] investigates the performance of LLMs on this task, which it frames as non-taxonomic relation extraction. Compared with taxonomy discovery, discussed in Section 2.2, relation learning appears to be more challenging. On a dataset constructed from UMLS [16] containing 53 non-taxonomic relation types, zero-shot performance is limited, with a maximum F1 score of 49.5. Fine-tuning yields only modest improvements, increasing the F1 score to 53.1. Mai et al. [56] consider a more constrained generation setting by extending hypernymy relations to holonymy, that is, *partOf* relations.

Bakker et al. [11] extends the evaluation to additional relation types, including *disjointWith*,

subPropertyOf, property domain, and property range. Their results show that extracting *subClassOf* relations is comparatively easier, achieving an F1 score of 0.359, whereas predicting domain and range relations is substantially more difficult, with F1 scores below 0.04.

2.4 Ontology Population

An ontology is typically divided into two components: a *TBox* and an *ABox*. The TBox captures conceptual knowledge, including class definitions and hierarchical structures, whereas the ABox contains assertions about individual instances. These assertions include class membership statements, such as $C(a)$, and relation assertions, such as $r(a, b)$, which specify relationships between instances a and b. In this section, we focus on ontology population, the task of automatically generating ABox assertions, typically assuming that the TBox has already been defined. Approaches that operate without an explicit TBox are generally studied under the broader task of *knowledge graph completion*, where an assertion $r(a, b)$ is represented as a triple (a, r, b), and are therefore outside the scope of this survey.

Most existing approaches perform ontology population by extracting knowledge from external sources. For example, SPIRES [20] introduces a schema-guided extraction framework in which a knowledge schema, consisting of classes, attributes, and value constraints, is incorporated into prompt instructions. Given textual documents and the schema specification, an LLM iteratively extracts facts and outputs them in a structured format. The approach achieves performance comparable to domain-specific fine-tuned models; for example, it attains an F1-score of 43.8 on the chemical–disease relation extraction task, compared with 44.98 achieved by BioGPT. Similarly, Norouzi et al. [62] employ a schema-guided extraction process but augment it with retrieval-augmented generation (RAG) to reduce irrelevant context. Specifically, documents are first segmented into chunks, and a summary of the ontology schema is generated. A RAG system then retrieves potentially relevant chunks, from which candidate ABox assertions are extracted. Structured decomposition [75] further strengthens this schema-guided direction by injecting TBox and SWRL rules into the extraction process and then using symbolic checking to verify or repair generated ABox assertions.

Other studies generate ABox assertions directly from the knowledge encoded within LLMs, without relying on external documents. For instance, Ayad and Khelifa [9] investigate ontology population in the asbestos domain. In this approach, the ontology schema is transformed into a CSV-based representation and provided to the LLM as input. The model is then prompted to populate the ontology using its parametric knowledge, generating ABox assertions that are subsequently converted back into a standard ontology format.

2.5 Ontology Learning Methods

Table 1 shows that LLM-based ontology learning is currently dominated by prompting-based methods. Prompt-only approaches constitute the largest group and have been applied to taxonomy completion, taxonomy construction, relationship learning, and ontology population [99, 80, 97, 87, 42, 20, 11]. Fine-tuning is also widely used, either as a standalone strategy [54, 59, 53, 90] or in combination with prompting [32, 22, 91]. By contrast, explicit RAG remains relatively rare and is mainly employed for ontology population or evidence-grounded ontology construction [62].

This distribution reflects the characteristics of ontology learning tasks. Many such tasks can be naturally formulated as relation judgment, concept placement, path generation, or schema-guided extraction, making prompting an intuitive and low-cost solution. Prompting can also exploit label names, definitions, and schema descriptions, aligning well with the few-shot and instruction-following capabilities of LLMs [18, 67]. However, prompting alone is often brittle in specialized

domains or when richer relation types are required. In these settings, domain adaptation through fine-tuning or ontology-derived supervision is often necessary [53, 59, 11]. RAG, by contrast, is most useful when ontology elements must be grounded in external evidence, such as documents, tables, or literature collections [62].

2.6 Discussion and Perspectives

The studies reviewed above demonstrate the substantial potential of LLMs for ontology learning. LLMs can flexibly interpret labels, definitions, documents, tables, and schemas, and can serve as semantic generators, candidate rankers, or schema-aware extractors. However, they still face important limitations when they are expected to produce complete and formally reliable ontologies without additional structural guidance or verification. We identify several promising directions for future research.

- **From generation to generate–rank–verify pipelines.** Many successful approaches already move beyond pure one-shot generation. For example, Chain-of-Layer filters LLM-generated edges, OLLM merges and prunes document-level paths, ICON separates missing-concept discovery from concept insertion, and ontology-population systems use schema summaries or retrieval-augmented generation to constrain the extraction context. These results suggest that future ontology learning systems should be designed as multi-stage pipelines. In such pipelines, LLMs can generate or verbalize candidate concepts, relations, and axioms; embedding models can rank and contextualize these candidates; and ontology reasoners or constraint checkers can verify whether the resulting hierarchy and assertions are structurally and logically acceptable.
- **Integration with hierarchy embeddings.** Hierarchy embedding methods, including box embeddings and hyperbolic representations, provide a promising direction for taxonomy construction and completion because they model asymmetric and transitive hierarchical structures more directly than text-only encoders. Although box embeddings have already been used for taxonomy completion, they are rarely integrated deeply with LLM-based methods. A promising research direction is to use LLMs to provide semantic evidence, explanations, and candidate parent or child concepts, while hierarchy embeddings provide global structural priors and uncertainty estimates for concept placement.
- **Domain adaptation and low-resource transfer.** Comparisons across LLMs4OL, the approach of OntoTune, TaxoLLaMA, and TaxoPro indicate that general-purpose LLMs can perform reasonably well in broad or lexical domains, but their performance often degrades substantially in specialized domains. Fine-tuning, ontology-driven self-training, and cross-domain transfer can improve performance, yet current evaluations remain fragmented across datasets, ontology types, and task formulations. Future work should therefore study how LLMs can be adapted using ontology-derived supervision, while evaluating not only downstream F1 scores but also whether the adapted models preserve domain constraints, avoid spurious general-domain associations, and transfer robustly to unseen ontology modules.
- **Toward richer formal semantics.** Current LLM-based approaches mainly focus on concepts, taxonomic relations, and simple relational assertions. In practice, ontologies often require richer semantics, including equivalence, disjointness, property hierarchies, domain and range restrictions, existential restrictions, and cardinality constraints. The results of OntoAxiom and recent work on complex concept placement suggest that these axiom types remain substantially more challenging than subclass prediction. Future systems should therefore move beyond direct axiom generation and develop reasoner-in-the-loop methods, in which LLMs propose candidate axioms and formal tools assess their consistency, entailment impact, and compatibility with existing modelling patterns.

Table 2 Summary of Papers of LLM-based OM. Simple and Complex refers to whether the matching involves only named concepts or complex concepts.

Paper	Task	Method	Models
He et al. (He'23) [38]	Simple	LLM as Matcher	T5
OLaLa (Hertling'23) [40]	Simple	LLM as Matcher	SBERT
Norouzi et al. (Norouzi'23) [63]	Simple	LLM for End-to-End OM	
Zamazal et al. (Zamazal'24) [83]	Complex	LLM as Matcher	
LLMs4OM (Giglou'24) [33]	Simple	LLM as Matcher	
HybridOM (Totoian'24) [85]	Simple	LLM for Semantic Enrichment	
Agent-OM (Qiang'24) [69]	Simple	LLM as Agent	
Amini et al. (Amini'24) [3]	Complex	LLM for End-to-End OM	
Taboada et al. (Taboada'25) [84]	Simple	LLM as Matcher	SBERT
OntoAligner (Giglou'25) [34]	Simple	LLM as Matcher	SBERT
Hu et al. (Hu'25) [43]	Simple	LLM for Semantic Enrichment, LLM as Matcher	SBERT
Diallo et al. (Diallo'25) [27]	Simple	LLM as Matcher	
GenOM (Song'26) [81]	Simple	LLM as Matcher, LLM for Semantic Enrichment	
Lushnei et al. (Lushnei'26) [55]	Simple	LLM as Matcher	

Model icons: PLM; BERT BERT; SBERT SBERT; T5 T5/FLAN-T5; GPT/ChatGPT; LLaMA; Claude; Mistral; Qwen; Falcon/TII; Gemini; DeepSeek; Kimi; GLM GLM; Phi Phi; Aya Aya; InternLM InternLM; DLM Domain.

Overall, future ontology learning systems should not rely solely on free-form LLM generation. A more robust and scientifically convincing direction is to develop neuro-symbolic architectures in which LLMs contribute linguistic and domain knowledge, embedding models capture latent hierarchical structure, retrieval mechanisms provide grounded evidence, and formal reasoners enforce consistency, validity, and explainability.

3 Ontology Matching

Ontology matching (OM), also known as ontology alignment, aims to find correspondences between concepts with some specific relationship like equivalence and subsumption across two given ontologies for their integration and reuse. Such a correspondence between two concepts is usually called mapping. Researchers in the OM community mostly focus on mappings between named (atomic) concepts, but there is also an increasing attention to mappings involving complex concepts which are defined with logic constructs in Description Logic like conjunction, disjunction and existential restriction. For simplicity, in this paper we regard the former OM tasks as *simple matching*, and the latter OM tasks as *complex matching*.

Due to the emergent capabilities of text understanding and natural language inference, LLMs have been very widely investigated for OM as for ontology learning. As in previous OM research, most of the recent LLM-based research and systems aim at simple matching with the equivalence relationship between named concepts [38, 40, 63, 69, 33, 85, 84, 27, 43, 34, 81, 55], and only very few consider complex matching involving equivalence between named and complex concepts [3, 83]. The majority of all these works evaluate their methods using datasets from the Ontology

Alignment Evaluation Initiative (OAEI)[4] including the tracks of Conference by [63, 69], Bio-ML by [38, 33, 85, 84, 81, 43], Anatomy by [40, 33, 34, 55], Biodiversity and Ecology (biodiv) by [40, 33, 34, 43], Disease and Phenotype by [33, 34], Material Sciences and Engineering by [33, 34, 43, 43], Knowledge Graph by [40, 33, 34, 43] and Large BioMed by [55], as well as other OM datasets such as the GeoLink complex alignment benchmark [104] used by [3]. The real-world ontologies used for evaluation are mostly from the life science domains especially biomedicine but also include material, engineering, earth and environment. Different from ontology learning, the tasks of OM are quite homogeneous, and therefore this survey focuses on the categorisation of these LLM-based works from the dimension of methodology as in the following sub-section, where more details of each work will be discussed.

3.1 LLM-based OM Methods

Some early studies explore end-to-end OM via directly feed the ontology into LLM-powered conversation systems for output mappings, while most of the other studies combine LLMs with other technologies for composing some pipelines with different steps for OM. According to the role of LLM, this survey categorizes the LLM-based OM methods into four types: LLM for end-to-end OM, LLM as matcher, LLM for semantic enrichment, and LLM as agent.

LLM for End-to-End OM. Norouzi et al. [63] are one of the earliest explorations of using LLMs for OM, by directly feeding two ontologies into ChatGPT with a prompt composed of problem definition, ontology triples that are transformed into formatted text following the pattern Predicate(Subject, Object), and objective. After an exploration of prompt variants, this end-to-end method with ChatGPT (v4.0) achieves higher recall but much lower precision than many classic OM systems on the OAEI conference track. Besides the matching inaccuracy, the limited context length of ChatGPT will also impact the usage of such a conversational solution for larger ontologies. Amini et al. [3] test matching the GeoLink Base Ontology (GBO) with the GeoLink Modular Ontology (GMO) for mappings involving complex concepts on ChatGPT. They first upload GMO' .ttl file, and then send a set of GBO concepts and ask for providing related pieces in GMO. They also provide relevant module information in GMO and use chain-of-though prompt for higher precision and recall over the baseline, but the quantitative results are still moderate.

LLM as Matcher. Instead of asking LLMs to deal with all the processing and directly output mappings of two original ontologies, most other studies prefer to use LLMs for matching concepts, combining them with other OM technologies or augmenting existing OM methods, for more applicable and more accurate pipeline style OM systems. These works often first extract a limited number of candidate concept correspondences by e.g., the lexical similarity between concept labels, and the Cosine similarity between concept embeddings based on Sentence-BERT, and then apply LLMs for assessing the equivalence of these correspondences, using different prompts based on the concepts' meta information like labels and synonyms, and their context like parent and child concepts either with or without few-shot examples. OLaLa [40] uses two LLM-based marking solutions: (i) binary decision which feeds one correspondence to the LLM and calculates the score as a positive mapping according to the appearance of positive terms like "true" and "yes" in the LLM output, and (ii) multiple choice decision which feeds one source concept and its multiple candidate target concepts into the LLM, and asks the LLM to select one target concept or returns "none". LLMs4OM [33], OntoAligner [34] which adopts LLMs4OM use a similar binary decision solution as OLaLa. He et al. [38], Hu et al. [43] and GenOM [81] also adopt binary decision but prompt the LLM for only a "yes" or "no" answer, and calculate the

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

score based on the output logits of the "yes" and "no" tokens in the final layer of the LLM. Note that the mapping scores enable the ranking of candidate correspondences, allowing more flexible post-processing and adoption to different system requirements like a bias to precision or recall.

LLM-based matching often leads to higher accuracy, but is more costly, especially by calling APIs of proprietary LLMs. Both Taboada et al. [84] and Lushnei et al. [55] further optimise the pipeline for reducing the time of using LLMs beyond normal solutions of calculating candidate concept correspondences. In particular, [84] selects those bidirectional correspondences whose source and target concepts rank each other in the top position based on embedding-based similarity, while [55] only uses the LLM to validate the correspondences that the classic OM system LogMap is very uncertain. Both [84] and [55] prompt LLM to directly answer a binary label for each correspondence without calculating the score as in [83, 27].

LLM for Semantic Enrichment. Inspired the text generation capability and large knowledge memorisation by LLMs, some OM studies use LLMs to extend the concept information before taking the normal OM steps like candidate retrieval and concept matching. Both HybridOM [85] and GenOM [81] prompt an LLM to generate textual descriptions of concepts in biomedical ontologies. HybridOM inputs a concept, which is a medical product, a medical disease or an anatomical human body concept, and asks the LLM to generate descriptions of some attributes selected from definition, chemical structure, cause and so on. Instead, GenOM feeds the concept as well its meta information and context like parent concepts to generate one single description as its definition. Hu et al. [43] introduce an LLM prompting-based infusion module that first checks the clarity and interpretability of a concept label and then generates a more informative reformulation for the concept with minimal and plausible changes if the label is found to be overly brief or ambiguous.

LLM as Agent. Besides knowledge memorisation, natural language reasoning and generation, recent LLMs also show impressive performance on planning, searching and calling external tools, which are all known as important features of agentic AI. Agent-OM [69] uses an LLM to automatically combine and use different tools for OM with a shared memory. It plans and calls different retrieval tools such as metadata retriever, semantic retriever and embedder, as well as different matching tools such as similarity search, metadata matcher, matching validator and merger.

3.2 Discussion and Perspectives

Like many areas in data and knowledge engineering, the adoption of LLMs in OM has revolutionized the existing methods, leading to substantial performance gains. The current LLM-based OM studies primarily leverage LLMs to improve the matching of named concepts, and give rise to two emerging OM paradigms: conversational (or dialogue-based) systems and generative approaches for enriched textual concept representation. Based on the above review of these studies, we have the following discussion and perspectives on both task and methodology:

- **Complex Matching**. The current exploration is limited to simple matching between named concepts, with only two preliminary studies on matching involving complex concepts — [83] which tests a zero-shot prompt on the small OAEI conference ontologies with only 8 positive mappings, and [3] which tests the conversational approach with zero-shot, few-shot and chain-of-thought prompts on 108 mappings between two GeoLink ontologies. LLM-based OM for complex matching lacks systematic evaluation on more data and more approaches of applying LLMs.
- **None Equivalence Matching**. Matching concepts with other relationships beyond equivalence such as subsumption, as benchmarked by OAEI tracks of Bio-ML [39] and Beyond-

Equivalence [51], has not been considered by these existing LLM-based OM studies. Given the success of LLMs in natural language inference, as demonstrated by studies on ontology subsumption inference and complex reasoning reviewed in this survey, we believe that LLMs have the potential to achieve even more performance gains over existing methods on none equivalence matching than on equivalence matching. Note that real-world ontology integration and reuse require not only equivalence matching between named concepts, but also more complex matching involving complex relationships and/or complex concepts, as well as other tasks like concept resolution where potential conflicts need to be addressed.

- **LLM Training for Adaption**. Ontologies often represent domain-specific knowledge, while the LLMs adopted in the existing OM studies are all of general purpose, lacking an exploration of LLM training for an adaption to both domain-specific knowledge and reasoning.
- **LLM for Agentic AI**. Although there is one OM study that explores LLMs for calling tools (i.e., Agent-OM [69]), more features of LLMs for agentic AI can be further investigated, such as memory utilisation, planning and task decomposition, think-and-act, multi-agent collaboration, coding and structured output generation. All of these capabilities can further enhance the role of LLMs in enabling more automated and accurate ontology integration and reuse.

4 Ontology-aware Complex Reasoning

This section examines complex reasoning tasks involving LLMs and ontologies that go beyond ontology learning and ontology matching. To structure the discussion, we divide this section into two parts, each corresponding to one of the following research questions:

- **RQ1:** How are ontologies used to support LLM-based reasoning and generation?
- **RQ2:** How are LLMs used to support ontology reasoning tasks?

The answers to these questions are presented in Sections 4.1 and 4.2, respectively. The surveyed papers are summarized in Table 3. Since the research tasks covered under this topic are diverse, we organize this part according to how ontologies are used. In Section 4.1, which addresses RQ1, we examine three main directions: 1. *ontology-augmented generation* (Section 4.1.1), where ontological knowledge or logical constraints are integrated into LLM generation through fine-tuning or training; 2. *ontology-augmented retrieval* (Section 4.1.2), where ontologies are used as schemas, rules, or external knowledge retrieval and directly provided as input to LLMs; and 3.*neuro-symbolic pipelines* (Section 4.1.3), which integrate ontology-based reasoning processes or reasoning results into LLM-based systems.

In Section 4.3, we classify the methodologies for using LLMs into prompting, fine-tuning, retrieval-augmented generation (RAG), and their combinations. The overall classification of tasks and methods are summarized in Table 3.

4.1 Ontology for LLM Reasoning and LLM Generation

In the existing literature, three main approaches integrate ontologies into LLM reasoning and generation: ontology-augmented generation, ontology-guided retrieval, and neuro-symbolic pipelines.

4.1.1 Ontology-Augmented Generation

By ontology-augmented generation, we refer to approaches that integrate ontology-derived knowledge into model training, representation learning, or prompting so that LLMs produce more domain-grounded responses. For example, OntoTune [53] fine-tunes LLMs using query–response

Table 3 Summary of papers discussed in Complex Reasoning.

Paper	Task	Method	Models
PrOntoQA [77] (Saparov et al. 2022)	Ontology-style formal QA	Prompting	GPT
[12] (Baldazzi et al. 2023)	Enterprise-domain generation	Fine-tuning	T5
[74] (Ronzano et al. 2024)	Representation learning	Fine-tuning	BERT
[2] (Allemang et al. 2024)	Text-to-SPARQL QA repair	Prompting	GPT
[86] (Wang et al. 2024)	Evaluation on DL-Lite	Prompting	GPT LLaMA
[13] (Baldazzi et al. 2024)	Soft ontological reasoning	Prompting + RAG	LLaMA
ORAG [88] (Xiao et al. 2024)	Fine-grained entity typing	RAG	LLaMA GPT
OntoTune [53] (Liu et al. 2025)	Domain-specific QA	Fine-tuning	LLaMA
OntoURL [102] (Zhang et al. 2025)	Ontological understanding, reasoning, and learning	Prompting	Phi LLaMA Aya InternLM Mistral DLM
OG-RAG [78] (Sharma et al. 2025)	Ontology-grounded QA	RAG	GPT LLaMA
OL-KGC [36] (Guo et al. 2025)	KG completion	Prompting	GLM
[23] (Cruz et al. 2025)	RAG over ontology-enhanced KGs	RAG	GPT
ORT [50] (Liu et al. 2025)	KG question answering	RAG	GPT DeepSeek Qwen
[31] (Giglou et al. 2025)	Ontology-constrained generation	Prompting	LLaMA Qwen
CyberRAG [103] (Zhao et al. 2025)	Cybersecurity QA	Prompting + RAG	LLaMA Mistral Qwen
OntoRAG [17] (Bran et al. 2025)	Scientific discovery	RAG	Claude LLaMA
NeurOWL [94]	Abductive reasoning	Prompting	SBERT Qwen
[24] (Damiano et al. 2025)	OWL Lite reasoning	Prompting	LLaMA Qwen DeepSeek
[75] (Sadowski et al. 2026)	ABox extraction and validation	Prompting	GPT Claude Gemini Qwen DeepSeek Kimi
LLM4Proof [93] (Yang et al. 2026)	OWL proof generation	Prompting	GPT Qwen DeepSeek LLaMA Mistral
[1] (Ali et al. 2026)	QA on Clinical KG	Prompting	GPT DeepSeek
SCAIR [21] (Chaturvedi et al. 2026)	QA on Enterprise KG	Prompting + RAG	GPT

Model icons: PLM; BERT BERT; SBERT SBERT; T5 T5/FLAN-T5; GPT/ChatGPT; LLaMA; Claude; Mistral; Qwen; Falcon/TII; Gemini; DeepSeek; Kimi; GLM GLM; Phi Phi; Aya Aya; InternLM InternLM; DLM Domain.

pairs constructed from information contained in the target ontologies, such as definitions, hypernyms, and synonyms. These pairs are then used as training data for techniques including supervised fine-tuning (SFT) and direct preference optimization (DPO), with the goal of encouraging the model to generate responses that more accurately reflect ontology-derived knowledge. Through this process, the fine-tuned models are expected to acquire domain-specific ontological knowledge. Evaluations on medical-domain-specific QA datasets show that OntoTune achieves a 0.8–6.8% improvement in Mean Reciprocal Rank (MRR) over existing methods across various datasets. The results also indicate that such domain-specific improvements do not compromise the models' general capabilities.

Other studies enrich the training or representation space with ontology-derived information. Ronzano and Nanavati [74], for example, investigate ontology-enhanced representation learning, in which domain ontologies provide semantic information and support the construction of additional training samples to improve the quality of LLM-generated embeddings in the relevant domains. Specifically, these samples are derived from the ontology's "is-a" hierarchy, together with LLM-

generated definitions of the corresponding concepts.

Baldazzi et al. [12] adopt a related but more reasoning-oriented approach. They apply ontology-based reasoning with Vadalog to domain-specific enterprise data in order to derive additional facts that are not explicitly stated. They then fine-tune LLMs on training data that include both the inferred facts and the corresponding reasoning traces. Their proof-of-concept evaluation demonstrates the potential benefits of this approach.

4.1.2 Ontology-Augmented Retrieval

Ontology-augmented retrieval aims to retrieve relevant information, evidence, or reasoning paths that are either drawn directly from an ontology or constructed with its guidance, and then provide them as input to LLMs to support response generation. These approaches can be broadly divided into two categories:

1. Ontology-based data construction or enhancement. Sharma et al. [78] use an ontology as a schema (e.g., pre-defined entities types and attributes) to extract and formalize information from documents, and then use RAG to retrieve query-relevant evidence from the ontology-grounded representation to improve answer quality.

Furthermore, Cruz et al. [23] provide a more detailed comparison of ontology-enhanced RAG. Specifically, they compare standard RAG, the more advanced GraphRAG method, and ontology-enhanced RAG. In the ontology-enhanced setting, ontologies generated from text or induced from databases are used to enrich the knowledge graph, which is then employed as the retrieval source in subsequent RAG steps. Their results [23] show that incorporating both ontologies and the original text chunks can substantially reduce time costs (30%) while preserving nearly the same QA accuracy as GraphRAG. This efficiency gain arises because ontology extraction only needs to be performed once and does not need to be repeated whenever new documents are added, unlike in the GraphRAG setting.

Related work also applies ontology-enhanced generation beyond QA. For instance, Guo et al. [36] study ontology-enhanced knowledge graph completion, while OntoRAG [17] uses ontology retrieval to support scientific discovery tasks.

2. Ontology-guided data retrieval or search. Ontologies can also be used to guide the data retrieval or search process. ORAG [88] introduces an ontology-guided RAG system for fine-grained entity typing (FET), a task that assigns entities to their corresponding classes (i.e., concepts in ontologies). Specifically, ORAG first enriches the class ontology with additional information from external sources, such as definitions, synonyms, and class instances, thereby improving retrieval performance for a given query. After retrieving candidate classes, the system further searches for their parent and child nodes, and then provides all retrieved information to the LLM to support the final classification decision. Experiments on two constructed datasets—one focused on redox-active organic electrode materials and the other on online posts about Genshin Impact—show that ORAG consistently improves F1 scores by 10%–24% compared with a vanilla RAG system.

ORT [50] uses the ontology associated with a knowledge graph (KG) to guide the retrieval component of a RAG system. Specifically, given a query over KG, the method first identifies the class label of the target entity, as well as the class labels of related entities, which are referred to as condition labels. It then traces possible reasoning paths over the ontology starting from the target class and prunes these paths according to the condition labels. The authors focus on a simple form of ontology that can be represented as triples of classes and relations. For instance, $A \sqsubseteq B$ can be represented as $(A, isSubClassOf, B)$, while $A \sqsubseteq \exists r.B$ can be represented as (A, r, B). Accordingly, ontology-level reasoning paths can be viewed as paths composed of edges represented by such triples. Finally, the method extracts KG paths that satisfy the corresponding class-level reasoning

paths and feeds these paths into an LLM to generate the final answer. Experiments on commonly used datasets, such as WebQSP and CWQ, show that ORT achieves better performance than classical KG-RAG methods such as RoG.

Similarly, SCAIR [21] uses ontologies to guide the recursive search performed by an LLM agent over an underlying KG. In particular, SCAIR uses a schema—which can be viewed as a lightweight ontology consisting of entity types, relations, and attributes—to guide retrieval over the KG. Its approach adapts the standard LLM-agent framework ReAct [96] in two ways. First, it introduces a lightweight planning stage to identify consistent relation paths in the KG. Second, during the iterative agentic search process, the schema is used to filter and select possible subsequent paths, thereby constraining the search space and improving retrieval relevance.

4.1.3 Neuro-symbolic Pipelines

The central idea of neuro-symbolic pipelines is to address reasoning tasks expressed in natural language by first translating the query, context, or premises into a symbolic representation, and then deriving the answer using standard reasoners, solvers, or provers.

Representative neuro-symbolic systems include Logic-LM [68], LINC [65], and LogicLLaMA [95]. However, these systems mainly target general symbolic logic settings, in particular first-order logic or solver-based logical reasoning, rather than ontology-based formalisms considered in this survey. We therefore do not discuss them in detail; broader coverage can be found in existing surveys on logical reasoning with language models [49].

In the following, we restrict our discussion to works that are directly related to, or primarily focused on, ontology-based approaches.

1. Text2SPARQL. A first line of work incorporates ontologies by converting natural-language questions into structured SPARQL queries, so that answers are retrieved from an explicit knowledge base rather than generated directly by LLMs. Ali et al. [1] address question answering in a specific clinical domain with the aim of reducing LLM hallucinations. They construct a domain-specific Resource Description Framework (RDF) ontology and knowledge graph from clinical and hospital data collected across multiple Egyptian institutions. Their neuro-symbolic pipeline first translates user questions into SPARQL queries and then executes these queries over the RDF ontology to retrieve the corresponding answers. This substantially improves answer accuracy (from 52% with DeepSeek-R1 to 98%) and greatly reduces hallucinations (from approximately 63% with ChatGPT-4 and 48% with DeepSeek-R1 to only 1.7%).

Other studies in this category focus less on building the underlying ontology and more on improving the quality of generated SPARQL queries. Allemang and Sequeda [2] propose an ontology-guided error detection and repair framework. Their method first checks whether generated SPARQL queries conform to ontology-based constraints, and then feeds the corresponding error messages back into LLMs to repair invalid queries. This substantially improves query-generation accuracy (from 54% to 72%).

Jiang et al. [44] also study the text-to-SPARQL task, but incorporate ontology information directly into the prompt. They decompose query generation into two stages: first generating the query structure and then filling in the concrete content to obtain the final SPARQL query. This strategy achieves strong performance on some datasets, such as SciQA (up to 0.9919 F1), but shows limited performance on others, such as DBLP-QuAD (0.6619 F1).

CyberRAG [103] follows a similar ontology-informed prompting direction, but focusing on the sepcific cybersecurity education domain. Rather than using ontology reasoners to formally verify generated answers, CyberRAG injects ontology information directly into the LLM prompt through prompt learning.

2. Others. Baldazzi et al. [13] introduce a soft reasoning process to augment ontological reasoning with real-world semantic flexibility. They achieve this by mapping natural language sentences into the Vadalog [14] Datalog format and reasoning over a set of Vadalog rules. This process can be further improved by a RAG system that adds extra facts. It is worth noting that the Vadalog-based ontological reasoning procedure in this work differs from standard ontology reasoning tasks, such as subsumption inference. Instead, it focuses more on deriving new facts through a chase procedure [19].

Sadowski and Chudziak [75] investigate the extraction and generation of ABox data from documents by directly incorporating TBox rules into the prompt and applying SWRL rules [41] to populate and verify the final results. On average over all 33 model-task combinations, their method improves the F1 score by 0.046 compared with few-shot learning. In particular, the symbolic checking step plays an important role, contributing an additional 0.097 to the F1 score.

4.2 LLMs for Complex Ontology Reasoning

Recently, the analysis of the logical reasoning capabilities of Large Language Models (LLMs) has attracted substantial attention [49]. In this survey, we focus on the use of LLMs for standard ontology reasoning tasks. Most existing studies concentrate on evaluating the reasoning abilities of LLMs [77, 86, 102, 24, 93], while only a few attempt to develop new ontology reasoning methods based on LLMs [94].

1. Reasoning Ability Evaluations. Wang et al. [86] evaluate the capabilities of LLMs in understanding DL-Lite ontologies along multiple dimensions, including ontology content, ontology formats, and reasoning ability. Specifically, they consider two groups of reasoning tasks: (1) derivation tasks, including concept subsumption ($C \sqsubseteq D$), role subsumption ($r \sqsubseteq t$), and instance assertions ($C(a)$); and (2) consistency checking, which focuses on detecting logical conflicts. For example, the axioms $A \sqsubseteq B$ and $A \sqsubseteq \neg B$ imply that A must be an empty concept and therefore cannot have any instances. Their results show that LLMs perform well on subsumption tasks, but their performance can drop substantially as the dataset size increases. For instance, on larger datasets, no derived instance assertions were found. Wang et al. also show that LLMs can perform consistency checking to some extent, but still struggle with complex datasets. However, this work primarily focuses on analysing the correctness and completeness of the final answers, without providing a detailed evaluation of the reasoning process itself.

Other benchmarks broaden this evaluation perspective. PrOntoQA [77] evaluates formal reasoning in synthetic ontology-like QA settings, OntoURL [102] tests symbolic ontological understanding, reasoning, and learning, and Damiano and Orciuoli [24] focus on OWL Lite reasoning. Together, these benchmarks show that LLM-based ontology reasoning should be assessed not only by final-answer accuracy, but also by logical setting, ontology expressiveness, input representation, and robustness to incomplete or noisy knowledge.

In contrast to answer-only evaluations, LLM4Proof [93] proposes a benchmark for evaluating the proof-generation capabilities of LLMs over ontologies by examining the reasoning process in greater depth. The work highlights both the promise of LLMs for providing explanations grounded in rigorous logics and the remaining challenges in supporting robust reasoning under complex or imperfect conditions. The benchmark is based on the notion of justifications, namely inclusion-minimal subsets of an ontology that entail a given subsumption[5]. Specifically, Yang et al. [93] introduce three tasks of increasing complexity to evaluate proof abilities: (1) *extraction*,

[5] That is, if $\mathcal{M} \subseteq \mathcal{O}$ is an inclusion-minimal sub-ontology of a given ontology $\mathcal{O}$ such that $\mathcal{M} \models \alpha$, then $\mathcal{M}$ is a justification for α over $\mathcal{O}$.

which requires extracting justifications from a given set of axioms and a target subsumption; (2) *simplification*, which assesses whether LLMs can simplify complex axioms into a weaker form that is still sufficient for deriving the conclusion; and (3) *explanation*, which requires generating a step-by-step proof. Building on existing justification-computation algorithms, they further propose an automated dataset-generation method. By evaluating different LLMs on the generated dataset, they report several observations:

- relatively advanced LLMs, such as GPT-4o-mini and Qwen3-32B, achieve strong overall results but still struggle with complex cases;
- the representation format, i.e., whether axioms are expressed in a formal logical language or in natural language, appears to have limited impact on LLM performance; and
- LLM performance is sensitive to noise and incomplete data, with performance reductions of 40–50% observed in settings with substantial noise or missing axioms.

2. Abductive Reasoning. Here, we focus on abductive reasoning over ontologies with LLMs, a relatively underexplored area. For broader work on abductive reasoning with LLMs, please refer to [76] and [37].

In NeurOWL [94], the authors propose an LLM-based method for a generalized abductive reasoning task. Standard abductive reasoning aims to identify missing axioms that explain observed facts, and typically assumes a predefined set of candidate axioms from which the missing axioms can be selected. NeurOWL generalizes this setting in two ways. First, it allows the given conclusion to be a hypothesized statement rather than a confirmed observation; consequently, the conclusion may be incorrect and should be verified by the system. Second, it does not require a predefined set of candidate missing axioms. Instead, given an incomplete ontology and a target subsumption, the method attempts to identify a missing intermediate concept that naturally induces the missing axioms. For example, if the ontology cannot derive a correct subsumption $A \sqsubseteq B$, NeurOWL aims to find an intermediate concept C such that $A \sqsubseteq C \sqsubseteq B$ should hold, and then regards the corresponding bridge axioms, namely $A \sqsubseteq C$, $C \sqsubseteq B$, or both, as missing axioms.

To achieve this goal, NeurOWL leverages ontology embeddings to filter possible intermediate concepts C, and then uses LLMs to assess their correctness. It also develops a progressive reasoning process that exploits existing knowledge as much as possible when searching for suitable intermediate concepts. In evaluations on real-world ontologies, the method achieves high accuracy in determining the correctness of given subsumptions, with F1 scores ranging from 0.84 to 0.97 across different datasets. It also achieves strong performance in identifying correct missing axioms on simpler datasets, with F1 scores between 0.74 and 0.89. However, its performance decreases on more complex cases, where F1 scores drop to 0.46–0.49, and under stricter evaluation settings, where the F1 score can be as low as 0.176 when all predictions are required to be correct. This degradation is mainly attributed to the extremely large candidate space and the limited performance of embedding models in such settings.

4.3 Complex Reasoning Methods

As shown in Table 3, complex reasoning studies mainly use LLMs through prompting, RAG, or their combinations. Prompting is the most common mode: LLMs are asked to answer ontology-style questions, judge entailments or consistency, generate SPARQL/Datalog-like symbolic forms, repair erroneous queries, produce proof steps, validate extracted assertions, or assess candidates produced by embeddings or symbolic filters [77, 2, 86, 102, 94, 24, 75, 93, 1]. In these cases, the ontology is often included in the prompt or used before or after the LLM call, but the LLM usage method remains prompting.

Fine-tuning appears less frequently and is mainly used for domain or representation adaptation. Baldazzi et al. [12] fine-tune models on enterprise-domain data enriched by ontology-based reasoning, while OntoTune [53] uses ontology-derived supervision for domain-specific QA. Ronzano and Nanavati [74] similarly use ontology-derived signals for representation learning. These studies differ from prompt-only work because ontology knowledge is incorporated into model parameters or learned representations rather than only being supplied at inference time.

RAG is the other major LLM usage pattern, especially when the task requires grounding answers in external documents, ontology-enriched knowledge graphs, or ontology-constrained paths. Some works [88, 78, 50, 23, 17] retrieve ontology-relevant evidence before generation, while others [103, 21] combine prompting with retrieval or agentic search. The important distinction is between the LLM usage method and the auxiliary role of ontologies: ontologies may guide retrieval, constrain generation, or validate outputs, but the corresponding LLM method is still prompting, RAG, fine-tuning, or a combination of these methods. Overall, complex reasoning currently relies more on hybrid pipelines than on standalone LLM reasoning, with retrieval modules, embeddings, ontology constraints, and symbolic reasoners providing grounding and verification around the LLM.

4.4 Discussion and Perspectives

Recent studies show that ontologies can make LLM-based reasoning more grounded, and that LLMs can in turn provide useful semantic signals for ontology-related reasoning tasks. However, the evidence reviewed in this section also suggests a clear boundary: LLMs are helpful as generators, translators, retrievers, validators, and explanation assistants, but they do not yet provide the soundness, completeness, and traceability expected from formal ontology reasoning. We identify several challenges and future directions.

- **From answer accuracy to faithful reasoning processes.** Most evaluations still focus on final answers, while ontology reasoning requires knowing why an answer follows, which axioms are used, and whether each step is logically valid. Benchmarks such as DL-Lite evaluation, OntoURL, and OWL proof generation show that LLMs can handle some ontology syntax and simple entailments, but performance drops with larger ABoxes, harder logical patterns, noise, and incompleteness [86, 102, 93]. Future work should therefore evaluate proof traces, justifications, counterexamples, and repair actions rather than only final labels. A useful target is a reasoning assistant that can expose its evidence, produce checkable intermediate steps, and explicitly distinguish entailed facts from plausible but unverified statements.
- **Reasoner-in-the-loop architectures.** LLMs are strong at interpreting natural language questions, producing candidate axioms, and proposing intermediate concepts, while symbolic reasoners remain necessary for entailment checking, consistency detection, and proof validation. Existing work already points in this direction: ontology-based query repair checks generated SPARQL against ontology constraints [2], structured decomposition verifies extracted ABox assertions with TBox and SWRL rules [75], and NeurOWL combines embeddings and LLM judgments for abductive ontology reasoning. Future systems should make this division of labor explicit: LLMs propose, verbalize, and prioritize; reasoners verify, reject, explain, and repair.
- **Ontology-aware retrieval beyond adding context.** Ontology-enhanced RAG systems demonstrate that schemas, class hierarchies, and relation constraints can improve retrieval and reduce hallucination [78, 50, 1, 103]. However, many systems still treat ontologies mainly as extra context. A stronger direction is to use ontologies as active retrieval controllers: selecting candidate entity types, constraining relation paths, detecting missing evidence, and checking whether generated answers respect domain constraints. This also requires provenance-aware

retrieval, because an ontology-derived answer should indicate which concepts, relations, axioms, and source documents support it.

- **Reasoning under incompleteness and inconsistency.** Real ontologies and knowledge graphs are often incomplete, partially inconsistent, or evolving. Current LLM-based reasoning methods remain much stronger in clean deductive settings than in abductive reasoning, ontology repair, inconsistency-tolerant reasoning, or open-world explanation. Future work should study how LLMs can help generate candidate repairs, rank alternative explanations, and communicate uncertainty, while formal methods maintain clear semantics about what is entailed, assumed, contradicted, or merely plausible.

Overall, future work should move from "LLM as a reasoner" toward auditable neuro-symbolic reasoning systems. The most promising systems will combine the linguistic flexibility of LLMs, the grounding power of retrieval, the integration support of ontology matching, and the formal guarantees of ontology reasoners.

5 Conclusion

This survey reviewed recent work on ontology construction and reasoning with LLMs across three connected areas: ontology learning, ontology matching, and complex reasoning. Across these areas, a common pattern emerges. LLMs are most effective when they are used as semantic interfaces: they can verbalize sparse labels, generate candidate concepts or axioms, rank or validate difficult correspondences, translate natural language into structured queries, and provide explanations that are easier for users to inspect. At the same time, they are much less reliable when they are expected to produce complete, consistent, and reusable ontological artifacts without external structure.

The central challenge is therefore not whether LLMs should replace ontology reasoning or engineering methods, but how they should be combined with them. Ontology learning still struggles with noisy generation, limited domain adaptation, and evaluation beyond local edge accuracy. Ontology matching remains concentrated on equivalence between named concepts, while complex matching, non-equivalence relations, conflict resolution, and scalable validation are still underexplored. Complex reasoning shows the strongest need for formal guarantees: LLMs can propose reasoning steps or missing knowledge, but symbolic reasoners are still required to check entailment, consistency, and proof correctness.

A promising direction for future work may be to explore auditable generate–rank–verify architectures. In such systems, LLMs could be used to generate candidate outputs and natural-language explanations, while retrieval modules provide supporting evidence, embedding and matching models rank alternatives, and ontology reasoners help enforce explicit constraints. This line of work would also benefit from benchmarks that assess not only task performance, but also process faithfulness, provenance, robustness to incomplete or inconsistent ontologies, and downstream usefulness. A potentially valuable avenue is the joint development of ontologies, knowledge graphs, alignments, and RAG systems, where learned or matched ontologies can inform retrieval and reasoning, and reasoning outcomes may help reveal missing concepts, weak mappings, or opportunities for ontology repair.

In summary, LLMs bring powerful linguistic and commonsense capabilities to ontology construction and reasoning, but their value is greatest when paired with the explicit semantics, constraints, and verification mechanisms of ontologies. The next stage of research should focus on building reliable, explainable, and maintainable neuro-symbolic systems rather than treating LLMs as standalone ontology engineers or reasoners.

References

1 Mohamed Ali, Zaki Taha, and Mohamed Mabrouk Morsey. Ontology-grounded knowledge graphs for mitigating hallucinations in large language models for clinical question answering. *Journal of Biomedical Informatics*, 175:104993, 2026. doi:10.1016/j.jbi.2026.104993.

2 Dean Allemang and Juan Sequeda. Increasing the accuracy of LLM question-answering systems with ontologies. In *International Workshop on the Semantic Web*, page 324–339, Berlin, Heidelberg, 2024. Springer-Verlag. doi:10.1007/978-3-031-77847-6_18.

3 Reihaneh Amini, Sanaz Saki Norouzi, Pascal Hitzler, and Reza Amini. Towards complex ontology alignment using large language models. In Sanju Tiwari, Boris Villazón-Terrazas, Fernando Ortiz-Rodríguez, and Soror Sahri, editors, *Iberoamerican Conference on Knowledge Graphs and Semantic Web*, volume 15459 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2024. doi:10.1007/978-3-031-81221-7_2.

4 Anthropic. Model system cards. https://www.anthropic.com/system-cards, 2026.

5 Pauline Armary, Cheikh Brahim El Vaigh, Ouassila Labbani-Narsis, and Christophe Nicolle. Ontology learning towards expressiveness: A survey. *Computer Science Review*, 56:100693, 2025. doi:10.1016/j.cosrev.2024.100693.

6 Ines Arous, Ljiljana Dolamic, and Philippe Cudré-Mauroux. Taxocomplete: Self-supervised taxonomy completion leveraging position-enhanced semantic matching. In *The Web Conference*, pages 2509–2518. ACM, 2023. doi:10.1145/3543507.3583342.

7 Michael Ashburner, C. Ball, J. Blake, D. Botstein, Heather Butler, J. Cherry, A. P. Davis, K. Dolinski, S. Dwight, J. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000. doi:10.1038/75556.

8 Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. A survey of ontology learning techniques and applications. *Database J. Biol. Databases Curation*, 2018:bay101, 2018. doi:10.1093/database/bay101.

9 Sarah Ayad and Lydia Khelifa Chibout. Ontology population with large language models (llms): A case study on asbestos ontology. *Appl. Ontology*, 20(1):89–106, 2025. doi:10.1177/15705838251334528.

10 F. Baader, Deborah L. McGuinness, Daniele Nardi, and P. Patel-Schneider. *The Description Logic handbook: Theory, Implementation and Applications*. Cambridge university press, 2003.

11 Roos M. Bakker, Daan L. Di Scala, Maaike H. T. de Boer, and Stephan A. Raaijmakers. Ontology learning with LLMs: A benchmark study on axiom identification. *arXiv.org*, abs/2512.05594, 2025. URL: https://arxiv.org/abs/2512.05594, doi:10.48550/arXiv.2512.05594.

12 Teodoro Baldazzi, Luigi Bellomarini, Stefano Ceri, Andrea Colombo, Andrea Gentili, and Emanuel Sallinger. Fine-tuning large enterprise language models via ontological reasoning. In Anna Fensel, Ana Ozaki, Dumitru Roman, and Ahmet Soylu, editors, *RuleML+RR*, volume 14244 of *Lecture Notes in Computer Science*, pages 86–94. Springer Nature Switzerland, 2023. URL: https://doi.org/10.1007/978-3-031-45072-3_6, doi:10.48550/arXiv.2306.10723.

13 Teodoro Baldazzi, Davide Benedetto, Luigi Bellomarini, Emanuel Sallinger, and Adriano Vlad. Softening ontological reasoning with large language models. *RuleML+RR*, 2024.

14 Luigi Bellomarini, Davide Benedetto, Georg Gottlob, and Emanuel Sallinger. Vadalog: A modern architecture for automated reasoning with large knowledge graphs. *Information Systems*, 105:101528, 2020. doi:10.1016/j.is.2020.101528.

15 Chris Biemann. Ontology learning from text: A survey of methods. *Ldv Forum*, 20(2):75–93, 2005. doi:10.21248/jlcl.20.2005.76.

16 Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Res.*, 32(90001):267D–270, 2004. doi:10.1093/nar/gkh061.

17 Andres M Bran, Alexandru Oarga, Matthew Hart, Magdalena Lederbauer, and Philippe Schwaller. Ontology-retrieval augmented generation for scientific discovery. https://openreview.net/forum?id=DbZDbg2z9q, 2025. OpenReview.

18 Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

19 A. Calí, G. Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *Sistemi Evoluti per Basi di Dati*, pages 77–86. ACM, 2009. doi:10.1145/1559795.1559809.

20 J. Caufield, Harshad B. Hegde, Vincent Emonet, N. Harris, marcin p. joachimiak, N. Matentzoglu, Hyeongsik Kim, S. Moxon, J. Reese, M. Haendel, et al. Structured prompt interrogation and recursive extraction of semantics (SPIRES): A method for populating knowledge bases using zero-shot learning. *Bioinform.*, 40(3):btae104, 2023. doi:10.1093/bioinformatics/btae104.

21 Prateek Chaturvedi, Yuqicheng Zhu, Hongkuan Zhou, Dongzhuoran Zhou, Yunjie He, Steffen Staab, Fei Du, Jie Tang, and Evgeny Kharlamov. Scair: Schema-conditioned agentic iterative reasoning for enterprise knowledge graphs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pages 1089–1104, 2026.

22 Boqi Chen, Fan Yi, and Dániel Varró. Prompting or fine-tuning? A comparative study of large language models for taxonomy construction. In *2023 ACM/IEEE International Conference on Model*

Driven Engineering Languages and Systems Companion (MODELS-C), pages 588–596. IEEE, 2023. doi:10.1109/MODELS-C59198.2023.00097.

23 T. Cruz, Bernardo Tavares, and Francisco Belo. Ontology learning and knowledge graph construction: A comparison of approaches and their impact on RAG performance. *arXiv.org*, 2025. URL: https://arxiv.org/pdf/2511.05991, doi:10.48550/arXiv.2511.05991.

24 Emanuele Damiano and Francesco Orciuoli. Evaluating large language models on OWL lite reasoning. In *International Conference on Semantic Systems*, CEUR Workshop Proceedings, 2025. URL: https://ceur-ws.org/Vol-4064/kgnesy-paper1.pdf.

25 DeepSeek-AI. DeepSeek-V4 technical report. https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf, April 2026.

26 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

27 Abdoulaye Diallo, Claudia d'Amato, and Mouhamadou Thiam. Ontology alignment validation using llm and kg. In *OM@ISWC*, 2025.

28 Hang Dong, Jiaoyan Chen, Yuan He, Yongsheng Gao, and Ian Horrocks. A language model based framework for new concept placement in ontologies. In Albert Meroño-Peñuela, Anastasia Dimou, Raphaël Troncy, Olaf Hartig, Maribel Acosta, Mehwish Alam, Heiko Paulheim, and Pasquale Lisena, editors, *Extended Semantic Web Conference*, volume 14664 of *Lecture Notes in Computer Science*, pages 79–99. Springer Nature Switzerland, 2024. URL: https://doi.org/10.1007/978-3-031-60626-7_5, doi:10.48550/arXiv.2402.17897.

29 Kevin P. Donnelly. SNOMED-CT: The advanced terminology and coding system for ehealth. *Studies in Health Technology and Informatics*, 121:279–290, 2006.

30 Maurice Funk, Simon Hosemann, Jean Christoph Jung, and Carsten Lutz. Towards ontology construction with language models. In Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, and Jeff Z. Pan, editors, *KBC-LM/LM-KBC@ISWC*, volume 3577 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3577/paper16.pdf, doi:10.48550/arXiv.2309.09898.

31 Hamed Babaei Giglou, Simon Burbach, Francesco Compagno, Muhammad Ismail, Gunjan Singh, and Sebastian Rudolph. Beyond statistical parroting: Hard-coding truth into llms via ontologies. 2025.

32 Hamed Babaei Giglou, J. D'Souza, and S. Auer. LLMs4OL: Large language models for ontology learning. In *International Workshop on the Semantic Web*, volume 14265 of *Lecture Notes in Computer Science*, pages 408–427. Springer Nature Switzerland, 2023. doi:10.48550/arXiv.2307.16648.

33 Hamed Babaei Giglou, Jennifer D'Souza, Felix Engel, and Sören Auer. Llms4om: Matching ontologies with large language models. In Albert Meroño-Peñuela, Óscar Corcho, Paul Groth, Elena Simperl, Valentina Tamma, Andrea Giovanni Nuzzolese, María Poveda-Villalón, Marta Sabou, Valentina Presutti, Irene Celino, Artem Revenko, Joe Raad, Bruno Sartini, and Pasquale Lisena, editors, *ESWC Satellite Events*, volume 15344 of *Lecture Notes in Computer Science*, pages 25–35. Springer, 2024. URL: https://doi.org/10.1007/978-3-031-78952-6_3, doi:10.48550/arXiv.2404.10317.

34 Hamed Babaei Giglou, Jennifer D'Souza, Oliver Karras, and Sören Auer. Ontoaligner: A comprehensive modular and robust python toolkit for ontology alignment. In Edward Curry, Maribel Acosta, María Poveda-Villalón, Marieke van Erp, Adegboyega K. Ojo, Katja Hose, Cogan Shimizu, and Pasquale Lisena, editors, *Extended Semantic Web Conference*, volume 15719 of *Lecture Notes in Computer Science*, pages 174–191. Springer Nature Switzerland, 2025. URL: https://doi.org/10.1007/978-3-031-94578-6_10, doi:10.48550/arXiv.2503.21902.

35 Google DeepMind. Gemini 3.1 Pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, February 2026.

36 Wenbin Guo, Xin Wang, Jiaoyan Chen, Zhao Li, and Zirui Chen. Ontology-enhanced knowledge graph completion using large language models. In *International Workshop on the Semantic Web*, pages 61–80. Springer, 2025. doi:10.48550/arXiv.2507.20643.

37 Kaiyu He and Zhiyu Chen. From reasoning to learning: A survey on hypothesis discovery and rule learning with large language models. *arXiv preprint arXiv:2505.21935*, 2025.

38 Yuan He, Jiaoyan Chen, Hang Dong, and Ian Horrocks. Exploring large language models for ontology alignment. In Irini Fundulaki, Kouji Kozaki, Daniel Garijo, and José Manuél Gómez-Pérez, editors, *International Workshop on the Semantic Web*, volume 3632 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3632/ISWC2023_paper_427.pdf, doi:10.48550/arXiv.2309.07172.

39 Yuan He, Jiaoyan Chen, Hang Dong, Ernesto Jiménez-Ruiz, A. Hadian, and Ian Horrocks. Machine learning-friendly biomedical datasets for equivalence and subsumption ontology matching. In *International Workshop on the Semantic Web*, pages 575–591. Springer, Springer International Publishing, 2022. doi:10.48550/arXiv.2205.03447.

40 Sven Hertling and Heiko Paulheim. OLaLa: Ontology matching with large language models. In *International Conference on Knowledge Capture*, pages 131–139. ACM, 2023. URL: https://dl.acm.org/doi/10.1145/3587259.3627571, doi:10.1145/3587259.3627571.

41 Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A semantic web rule language combining OWL and RuleML. W3c member submission, World Wide Web Consortium, May 2004. URL: https://www.w3.org/submissions/SWRL/.

42 Chao-Chun Hsu, Erin Bransom, Jenna Sparks, Bailey Kuehl, Chenhao Tan, David Wadden, Lucy Lu Wang, and Aakanksha Naik. CHIME: LLM-assisted hierarchical organization of scientific studies for literature review support. In *Annual Meeting of the Association for Computational Linguistics*, pages 118–132. Association for Computational Linguistics, 2024. URL: https://aclanthology.org/2024.acl-long.314, doi:10.48550/arXiv.2407.16148.

43 Wenxin Hu and Ryutaro Ichise. From matching to retrieval: A new role for llms in ontology alignment. In *OM@ISWC*, 2025.

44 Longquan Jiang, Xi Yan, and Ricardo Usbeck. A structure and content prompt-based method for knowledge graph question answering over scholarly data. In *QALD/SemREC@ISWC*, 2023.

45 Minhao Jiang, Xiangchen Song, Jieyu Zhang, and Jiawei Han. Taxoenrich: Self-supervised taxonomy completion via structure-semantic representations. In *The Web Conference*, pages 925–934. ACM, 2022. doi:10.1145/3485447.3511935.

46 Song Jiang, Qiyue Yao, Qifan Wang, and Yizhou Sun. A single vector is not enough: Taxonomy expansion via box embeddings. In *The Web Conference*, pages 2467–2476. ACM, 2023. doi:10.1145/3543507.3583310.

47 Navya Martin Kollapally, James Geller, V. Keloth, Zhe He, and Julia Xu. Ontology enrichment using a large language model: Applying lexical, semantic, and knowledge network-based similarity for concept placement. *Journal of Biomedical Informatics*, 168:104865, 2025. doi:10.1016/j.jbi.2025.104865.

48 Jiayi Li, Daniel Garijo, and M. Poveda-Villalón. Large language models for ontology engineering: A systematic literature review. *Semantic Web*, 2026. URL: https://www.semantic-web-journal.net/system/files/swj4039.pdf.

49 Hanmeng Liu, Zhizhang Fu, Mengru Ding, Ruoxi Ning, Chaoli Zhang, Xiaozhang Liu, and Yue Zhang. Logical reasoning in large language models: A survey. *arXiv.org*, 2025. doi:10.48550/arXiv.2502.09100.

50 Runxuan Liu, Bei-Bei Luo, Jiaqi Li, Baoxin Wang, Ming Liu, Dayong Wu, Shijin Wang, and Bing Qin. Ontology-guided reverse thinking makes large language models stronger on knowledge graph question answering. In *Annual Meeting of the Association for Computational Linguistics*, pages 15269–15284. Association for Computational Linguistics, 2025. doi:10.48550/arXiv.2502.11491.

51 Xianhao Liu, Sven Hertling, and Michael R Hansen. Beyond equivalence: Benchmark datasets for ontology alignment. In *Iberoamerican Conference on Knowledge Graphs and Semantic Web*, pages 122–136. Springer, 2025. doi:10.1007/978-3-032-13109-6_9.

52 Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv.org*, arXiv:1907.11692, 2019. arXiv:1907.11692.

53 Zhiqiang Liu, Chengtao Gan, Junjie Wang, Yichi Zhang, Zhongpu Bo, Mengshu Sun, Huajun Chen, and Wen Zhang. Ontotune: Ontology-driven self-training for aligning large language models. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov, editors, *The Web Conference*, pages 119–133. ACM, 2025. doi:10.1145/3696410.3714816.

54 Andy Lo, Albert Q. Jiang, Wenda Li, and Mateja Jamnik. End-to-end ontology learning with large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Neural Information Processing Systems*, pages 87184–87225. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. URL: http://papers.nips.cc/paper_files/paper/2024/hash/9e89f068a62f6858c661a8abecf5bb0a-Abstract-Conference.html, doi:10.48550/arXiv.2410.23584.

55 Sviatoslav Lushnei, Dmytro Shumskyi, Severyn Shykula, E. Jiménez-Ruiz, and Artur d'Avila Garcez. Large language models as oracles for ontology alignment. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2026. URL: https://arxiv.org/abs/2508.08500, doi:10.48550/arXiv.2508.08500.

56 Huu Tan Mai, Cuong Xuan Chu, and Heiko Paulheim. Do llms really adapt to domains? an ontology learning perspective. In *International Semantic Web Conference*, pages 126–143. Springer, 2024.

57 Meta. Llama 4 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama4/MODEL_CARD.md, April 2025.

58 Sahil Mishra, Srinitish Srinivasan, Srikanta Bedathur, and Tanmoy Chakraborty. Taxobell: Gaussian box embeddings for self-supervised taxonomy expansion. In *arXiv.org*, pages 4326–4337, 2026. doi:10.1145/3774904.3792674.

59 Viktor Moskvoretskii, Ekaterina Neminova, Alina Lobanova, Alexander Panchenko, and Irina Nikishina. TaxoLLaMA: Wordnet-based model for solving multiple lexical semantic tasks. In *Annual Meeting of the Association for Computational Linguistics*, pages 2331–2350. Association for Computational Linguistics, 2024. URL: https://aclanthology.org/2024.acl-long.97, doi:10.18653/v1/2024.acl-long.127.

60 Irina Nikishina, Polina Chernomorchenko, Anastasiia Demidova, Alexander Panchenko, and Chris Biemann. Predicting terms in is-a relations with pre-trained transformers. In *International Joint Conference on Natural Language Processing*, pages 134–148. Association for Computational Linguistics, 2023. doi:10.18653/v1/2023.findings-ijcnlp.12.

61 Yuhang Niu, Hongyuan Xu, Ciyi Liu, Yanlong Wen, and Xiaojie Yuan. Contrastive representation learning for self-supervised taxonomy completion. In *International Joint Conference on Artificial Intelligence*, volume 8, pages 6442–6450, 2024. doi:10.24963/ijcai.2024/712.

62 Sanaz Saki Norouzi, Adrita Barua, Antrea Christou, Nikita Gautam, Andrew Eells, Pascal Hitzler, and Cogan Shimizu. Ontology population using llms. *arXiv.org*, abs/2411.01612, 2024. URL: https://doi.org/10.48550/arXiv.

2411.01612, arXiv:2411.01612, doi:10.48550/ARXIV.2411.01612.

63 Sanaz Saki Norouzi, Mohammad Saeid Mahdavinejad, and Pascal Hitzler. Conversational ontology alignment with ChatGPT. In *OM@ISWC*, 2023. doi:10.48550/arXiv.2308.09217.

64 Natasha Noy. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05 and SMI-2001-0880, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, 2001. URL: http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.

65 Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176. Association for Computational Linguistics, 2023. doi:10.18653/v1/2023.emnlp-main.313.

66 OpenAI. Introducing GPT-5.5. https://openai.com/index/introducing-gpt-5-5/, April 2026.

67 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Neural Information Processing Systems*, volume 35, pages 27730–27744. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html, doi:10.52202/068431-2011.

68 Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Conference on Empirical Methods in Natural Language Processing*, pages 3806–3824. Association for Computational Linguistics, 2023. doi:10.48550/arXiv.2305.12295.

69 Zhangcheng Qiang, Weiqing Wang, and Kerry Taylor. Agent-om: Leveraging LLM agents for ontology matching. *Proceedings of the VLDB Endowment*, 18(3):516–529, 2024. URL: https://www.vldb.org/pvldb/vol18/p516-qiang.pdf, doi:10.14778/3712221.3712222.

70 Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model. https://qwen.ai/blog?id=qwen3.6-27b, April 2026.

71 Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. https://qwen.ai/blog?id=qwen3.6-35b-a3b, April 2026.

72 Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

73 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2019. URL: http://jmlr.org/papers/v21/20-074.html.

74 Francesco Ronzano and Jay Nanavati. Towards ontology-enhanced representation learning for large language models. *arXiv.org*, abs/2405.20527, 2024. URL: https://doi.org/10.48550/arXiv.2405.20527, arXiv:2405.20527, doi:10.48550/ARXIV.2405.20527.

75 Albert Sadowski and Jaroslaw A. Chudziak. Structured decomposition for LLM reasoning: Cross-domain validation and semantic web integration. *arXiv.org*, abs/2601.01609, 2026. URL: https://doi.org/10.48550/arXiv.2601.01609, arXiv:2601.01609, doi:10.48550/ARXIV.2601.01609.

76 Moein Salimi, Shaygan Adim, Danial Parnian, Nima Alighardashi, Mahdi Jafari Siavoshani, and Mohammad Hossein Rohban. Wiring the'why': A unified taxonomy and survey of abductive reasoning in llms. *arXiv preprint arXiv:2604.08016*, 2026.

77 Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*, 2022.

78 Kartik Sharma, Peeyush Kumar, and Yunqing Li. Og-rag: ontology-grounded retrieval-augmented generation for large language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 32950–32969, 2025. doi:10.48550/arXiv.2412.15235.

79 Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *The Web Conference*, pages 486–497. ACM, 2020. doi:10.1145/3366423.3380132.

80 Jingchuan Shi, Hang Dong, Jiaoyan Chen, Zhe Wu, and Ian Horrocks. Taxonomy completion via implicit concept insertion. In *The Web Conference*, pages 2159–2169. ACM, 2024. doi:10.1145/3589334.3645584.

81 Yiping Song, Jiaoyan Chen, and Renate A Schmidt. Genom: ontology matching with description generation and large language models. *World Wide Web*, 29(3):29, 2026.

82 Yushi Sun, Hao Xin, Kai Sun, Y. Xu, Xiao Yang, Xin Dong, Nan Tang, and Lei Chen. Are large language models a good replacement of taxonomies? *Proceedings of the VLDB Endowment*, 17(11):2919–2932, 2024. URL: https://www.vldb.org/pvldb/vol17/p2919-sun.pdf, doi:10.48550/arXiv.2406.11131.

83 Ondrej Sváb-Zamazal. Towards pattern-based complex ontology matching using SPARQL and LLM. In *International Conference on Semantic Systems*, CEUR Workshop Proceedings, 2024. URL: https://arxiv.org/abs/2404.10329.

84 Maria Taboada, Diego Martinez, Mohammed Arideh, and Rosa Mosquera. Ontology matching with large language models and prioritized depth-first search. *Information Fusion*, 123:103254, 2025. doi:10.48550/arXiv.2501.11441.

85 Marius-Horatiu Totoian, Anca Marginean, Philipp Blohm, and Mir Nawab Hussain. HybridOM: Ontology matching using hybrid search. In *OM@ISWC*, 2024.

86 Keyu Wang, Guilin Qi, Jiaqi Li, and Songlin Zhai. Can large language models understand dl-lite ontologies? an empirical study. In *Conference on Empirical Methods in Natural Language Processing*, pages 2503–2519. Association for Computational Linguistics, 2024. doi:10.48550/arXiv.2406.17532.

87 Zhenyu Wu, Jiaoyan Chen, and Norman W. Paton. Taxonomy inference for tabular data using large language models. *Extended Semantic Web Conference*, abs/2503.21810:403–422, 2025. URL: https://arxiv.org/abs/2503.21810, doi:10.48550/arXiv.2503.21810.

88 Jinfeng Xiao, Linyi Ding, James Barry, Mohab Elkaref, Geeth de Mel, and Jiawei Han. Orag: Ontology-guided retrieval-augmented generation for theme-specific entity typing. In *First Conference on Language Modeling*, 2024.

89 Hongyuan Xu, Ciyi Liu, Yuhang Niu, Yunong Chen, Xiangrui Cai, Yanlong Wen, and Xiaojie Yuan. Tacoprompt: A collaborative multi-task prompt learning method for self-supervised taxonomy completion. In *Conference on Empirical Methods in Natural Language Processing*, pages 15804–15817. Association for Computational Linguistics, 2023. doi:10.18653/v1/2023.emnlp-main.979.

90 Hongyuan Xu, Yuhang Niu, Ciyi Liu, Yanlong Wen, and Xiaojie Yuan. Taxopro: A plug-in lora-based cross-domain method for low-resource taxonomy completion. *Transactions of the Association for Computational Linguistics*, 13:557–576, 2025. doi:10.1162/tacl_a_00755.

91 Hongyuan Xu, Yuhang Niu, Yanlong Wen, and Xiaojie Yuan. Compress and mix: Advancing efficient taxonomy completion with large language models. In *The Web Conference*, pages 4239–4249. ACM, 2025. doi:10.1145/3696410.3714690.

92 Wei Xue, Yongliang Shen, Wenqi Ren, Jietian Guo, Shiliang Pu, and Weiming Lu. Insert or attach: Taxonomy completion via box embedding. In *Annual Meeting of the Association for Computational Linguistics*, pages 3851–3863, 2023. doi:10.18653/v1/2024.acl-long.212.

93 Hui Yang, Jiaoyan Chen, and Uli Sattler. Large language model for owl proofs. In *Proceedings of the ACM Web Conference 2026*, WWW '26, page 3952–3963, New York, NY, USA, 2026. ACM. doi:10.1145/3774904.3792395.

94 Hui Yang, Jiaoyan Chen, Yiping Song, Renate Schmidt, and Wen Zhang. Neurowl: An llm-based neural-symbolic framework for incomplete owl ontology reasoning, 2026. URL: https://arxiv.org/abs/2607.15776, arXiv:2607.15776.

95 Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. Harnessing the power of large language models for natural language to first-order logic translation. In *Annual Meeting of the Association for Computational Linguistics*, 2023. doi:10.48550/arXiv.2305.15541.

96 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

97 Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Zhenwen Liang, Zhihan Zhang, and Meng Jiang. Chain-of-layer: Iteratively prompting large language models for taxonomy induction from limited examples. In *International Conference on Information and Knowledge Management*, pages 3093–3102. ACM, 2024. doi:10.1145/3627673.3679608.

98 Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Zhenyu Wu, Shangbin Feng, and Meng Jiang. Codetaxo: Enhancing taxonomy expansion with limited examples via code language prompts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Annual Meeting of the Association for Computational Linguistics*, volume ACL 2025 of *Findings of ACL*, pages 4131–4144. Association for Computational Linguistics, 2025. URL: https://aclanthology.org/2025.findings-acl.214/, doi:10.48550/arXiv.2408.09070.

99 Qingkai Zeng, Jinfeng Lin, Wenhao Yu, Jane Cleland-Huang, and Meng Jiang. Enhancing taxonomy completion with concept generation via fusing relational representations. In *Knowledge Discovery and Data Mining*, pages 2104–2113. ACM, 2021. doi:10.1145/3447548.3467308.

100 Jieyu Zhang, Xiangchen Song, Ying Zeng, Jiaze Chen, Jiaming Shen, Yuning Mao, and Lei Li. Taxonomy completion via triplet matching network. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 4662–4670. Association for the Advancement of Artificial Intelligence (AAAI), 2021. doi:10.1609/aaai.v35i5.16596.

101 Lin Zhang, Zhouhong Gu, Suhang Zheng, Tao Wang, Tianyu Li, Hongwei Feng, and Yanghua Xiao. Lite: Llm-impelled efficient taxonomy evaluation. *arXiv.org*, 2025. doi:10.48550/arXiv.2504.01369.

102 Xiao Zhang, Huiyuan Lai, Qianru Meng, and Johan Bos. Ontourl: A benchmark for evaluating large language models on symbolic ontological understanding, reasoning and learning. *arXiv.org*, abs/2505.11031, 2025. URL: https://doi.org/10.48550/arXiv.2505.11031, arXiv:2505.11031, doi:10.48550/ARXIV.2505.11031.

103 Chengshuai Zhao, Garima Agrawal, Tharindu Kumarage, Zhen Tan, Yuli Deng, Ying-Chih Chen, and Huan Liu. Ontology-aware rag for improved question-answering in cybersecurity education. In *BigData Congress [Services Society]*, pages 3161–3170. IEEE, 2025. doi:10.1109/BigData66926.2025.11401740.

104 Lu Zhou, Michelle Cheatham, Adila Krisnadhi, and Pascal Hitzler. Geolink data set: A complex alignment benchmark from real-world ontology. *Data Intelligence*, 2(3):353–378, 2020. doi:10.1162/dint_a_00054.