CEUR-WS.org/Vol-4064/kgnesy-paper1.pdf

Evaluating Large Language Models on OWL Lite Reasoning

Emanuele Damiano[1], Francesco Orciuoli[1]

[1]*Dipartimento di Scienze Aziendali - Management & Innovation Systems,*
Università degli Studi di Salerno, via Giovanni Paolo II, 132, Fisciano (SA), Italia

Abstract

This work investigates the capability of large language models (LLMs) to interpret OWL Lite ontologies and perform reasoning over them. We propose an evaluation framework based on the well-known LUBM ontology that is transformed into text and vectorized by using an embedding algorithm, enabling retrieval-augmented generation (RAG) to support query answering. Any previous knowledge of LLMs related to LUBM has been excluded by using adequate prompts, in order to rely exclusively on the information locally obtained through RAG. A set of 53 manually constructed queries is used to probe the models' ability to perform ontology-based inference aligned with the ontology axioms. Such queries vary in complexity (from level 1 to level 3) based on the number and depth of required logical inference operations. No local knowledge about OWL and ontology-based reasoning has been provided to the models; therefore, we are confident to evaluate emergent abilities in the realm of reading, interpreting, and reasoning on OWL Lite ontologies. The answers provided by the LLMs are compared against a gold standard to compute accuracy. Furthermore, we evaluate and compare the performance of different existing models within this setting to assess their relative effectiveness in OWL-based reasoning tasks. The obtained results offer interesting insights into the reasoning potential of LLMs when grounded in symbolic ontological knowledge.

Keywords

Large Language Models, Web Ontology Language (OWL), Ontology-based Reasoning, Neurosymbolic Computing

1. Introduction and Motivations

Large Language Models (LLMs) are increasingly popular tools capable of sophisticated natural language processing, demonstrating their potential in diverse applications, including complex code generation and combinatorial problem-solving [1]. In this context, recent studies have sought to integrate LLMs with symbolic reasoners [2] to develop more robust and capable intelligent systems under the framework of Neurosymbolic Computing [3]. The central question addressed by this research is the degree to which LLMs can effectively process and leverage structured knowledge represented in formal ontologies, as well as how various well-known LLMs accessible from platforms like Groq, Ollama, and others compare in this capability. To this end, we delve into the abilities of LLMs in reading, interpreting, and reasoning over OWL ontologies, which are foundational for semantic web technologies and knowledge representation [4]. Understanding how LLMs interpret and reason over structured knowledge, such as OWL ontologies, is a fundamental challenge at the intersection of symbolic and neural AI. OWL ontologies represent formal, machine-readable semantics that underpin critical domains like healthcare, finance, legal systems, and scientific research. Demonstrating that LLMs can perform correct inferences over such ontologies—especially under constrained setups like Retrieval-Augmented Generation (RAG)—is not only a technical milestone but a conceptual leap toward integrating symbolic reasoning with neural models[5]. This has direct implications for neurosymbolic computing, where the goal is to combine the robustness of learning with the rigor of logic [6]. Moreover, proving that LLMs can act as semantic reasoners supports the development of more explainable AI systems, as ontological reasoning is inherently interpretable [7]. It also contributes to Generative eXplainable AI

The Second Workshop on Knowledge Graphs and Neurosymbolic AI (KG-NeSy), co-located with SEMANTiCS'25: International Conference on Semantic Systems, September 3–5, 2025, Vienna, Austria

e.damiano7@studenti.unisa.it (E. Damiano); forciuoli@unisa.it (F. Orciuoli)
0009-0003-3864-8443 (E. Damiano); 0000-0001-6899-4396 (F. Orciuoli)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

(GenXAI) by showcasing how generative models can produce grounded, semantically valid responses rooted in structured knowledge[8]. Ultimately, this research can redefine how we build intelligent systems—moving from purely data-driven responses to semantically-aware and logically-consistent behavior, with transformative potential across knowledge-intensive applications. The paper presents a methodology to evaluate the aforementioned ability by leveraging LUBM[1], a well-known benchmark originally used to assess the performance of triple stores, and constructing a set of queries (written in natural language) with increasing difficulty levels that require LLMs to make inferences over the LUBM ontology in order to provide the correct answers. In this paradigm, our study focuses on OWL Lite, a simplified subset of the Web Ontology Language, chosen for its balance between expressiveness and simplicity, making it suitable for assessing the foundational understanding of ontologies by LLMs. Specifically, we provide a prompt constraining the models to consider only the LUBM ontology provided locally through RAG and any existing knowledge about OWL Lite and ontology-based reasoning. In this way, we can evaluate models' emergent abilities needed to read, interpret, and reason on OWL Lite ontologies, i.e., the ability of LLMs to act as reasoners. The evaluation activities focus on both quantitative and qualitative aspects. The quantitative evaluations are mainly based on accuracy measures when comparing LLMs' answers to the gold standard during query execution. The qualitative evaluations, on the other hand, involve analyzing the LLMs' reasoning processes and evaluating their plausibility and faithfulness [9] [10] by extracting and inspecting a self-evaluation from the LLMs' answers.

2. Related Works

Recent studies have explored the ability of Large Language Models (LLMs) to interpret and reason over ontological knowledge expressed in formal languages. Two notable directions are particularly relevant to our work: the evaluation of symbolic knowledge implicitly learned by LLMs and the assessment of their reasoning capabilities over structured ontologies.

Authors of [11] introduce ONTOLAMA, a benchmark designed to evaluate whether LLMs encode semantic subsumption relations derived from OWL ontologies. Their methodology probes whether concepts of the form $C \sqsubseteq D$ are entailed by the model when translated into natural language templates. Crucially, their approach relies on zero-shot evaluation without external knowledge access and focuses exclusively on subsumption. While insightful in measuring internalized semantic knowledge, it does not test the model's ability to reason dynamically over a structured ontology.

In contrast, the work [12] proposes a comprehensive evaluation of LLMs' understanding of DL-Lite ontologies, examining tasks such as syntax checking, concept and role subsumption, instance checking, property characteristics, and query answering. Their framework involves prompting LLMs directly with formal axioms and evaluating their reasoning behavior without any external knowledge retrieval. The study demonstrates that while LLMs can handle simple axiomatic patterns, their performance degrades significantly when transitivity or larger ABoxes are involved.

Our work differs significantly from both approaches. Rather than probing the LLM's internalized ontology knowledge or reasoning over embedded axioms, we leverage a retrieval-augmented generation (RAG) architecture to provide the model with access to an OWL Lite ontology—specifically the LUBM benchmark—in TURTLE (TTL) format[13]. We encode the ontology using dense text embeddings and enable the model to retrieve relevant axioms during inference. The reasoning is evaluated through a curated set of natural language queries, stratified by inferential complexity. This allows us to assess whether the LLM can simulate an OWL reasoner when given symbolic knowledge dynamically at inference time, rather than relying on prior training.

From a neurosymbolic AI perspective, our approach demonstrates the feasibility of grounding LLMs in external ontological structures to support symbolic reasoning, thus bridging the gap between neural text processing and logical inference. Unlike prior works, we evaluate operational reasoning using

---

[1] https://swat.cse.lehigh.edu/projects/lubm/

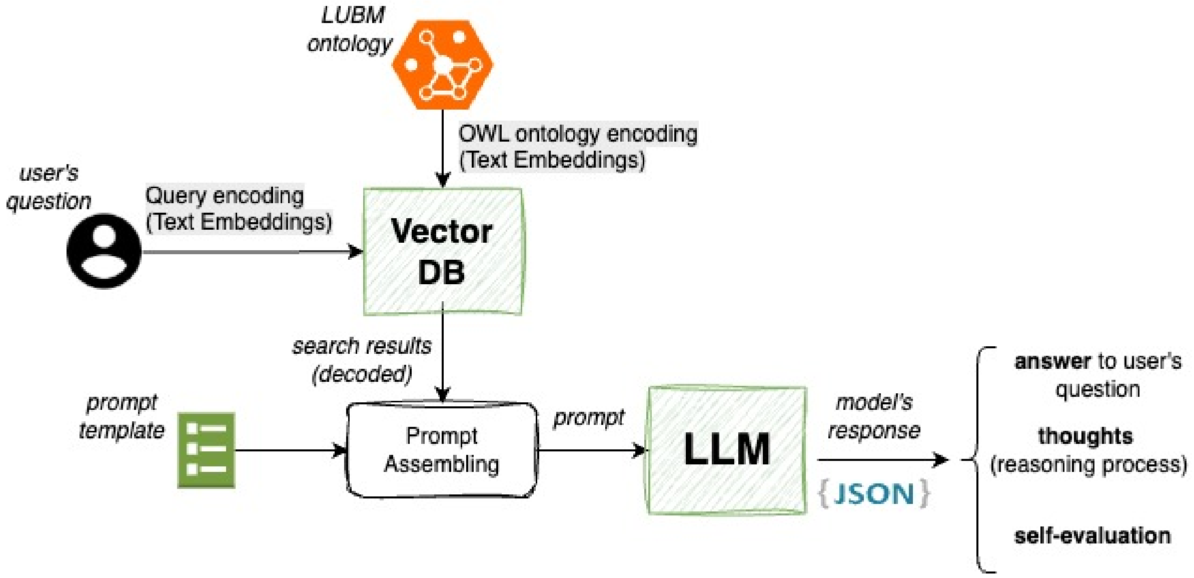


Figure 1: Sketch of the mechanism providing queries to LLMs.

realistic ontological content, reflecting practical challenges in knowledge-based systems and explainable AI.

3. Methodology

The methodology adopted to evaluate the ability of LLMs to read, understand, and make inferences over OWL Lite ontologies consists of two main phases. The first phase is to provide the OWL Lite ontology to the LLM and guide it to answer user's questions leveraging on both its knowledge about OWL Lite and the specific available OWL Lite ontology.

As reported in Fig. 1 and detailed in Fig. 2, an important aspect of the first phase involves prompt engineering. Such a prompt excludes previous knowledge about LUBM, which might conflict with the provided ontology, thereby enabling a fair evaluation of the models' ability to leverage both their understanding of OWL Lite, based on their prior knowledge, and their capability to apply such knowledge when working on specific OWL-based schemes like LUBM. The second phase consists of concretely assessing the aforementioned ability by comparing the performance of several models and analyzing quantitatively and qualitatively such performance. This is realized by executing a benchmark composed by the existing LUBM ontology and a new set of natural language questions (crafted by the authors of this work) requiring the execution of OWL Lite inferences over LUBM. The underlying idea is that if LLMs are able to answer questions requiring OWL Lite inferences, after limiting their knowledge to OWL Lite language and LUBM ontology, they demonstrate (within some limits) that they can read, understand, make inferences, and in some sense act as a symbolic reasoner. Such results are important for future works targeted at designing neurosymbolic systems.

3.1. OWL Lite

OWL Lite is a simplified sublanguage of the Web Ontology Language (OWL) designed for taxonomies and basic constraints, balancing expressiveness with computational tractability. It corresponds to the SHIF(D) description logic, enabling structured knowledge representation while maintaining decidability. The key inference types supported by OWL Lite are listed in Tab. 1. More expressive ontology languages (e.g., OWL 2) will be considered in further works.

Table 1
Key Inference Types Supported by OWL Lite

Inference Type	**Description**
Subsumption Reasoning	Determines subclass and subproperty relationships. Enables deduction of implicit class/property hierarchies using TBox reasoning (e.g., if $A \sqsubseteq B$ and $B \sqsubseteq C$, then $A \sqsubseteq C$)[5][6].
Equivalence Inference	Identifies when two classes or properties are equivalent, including handling of `owl:sameAs` for individuals, leading to entailment of all facts about one individual to all equivalents[1][2].
Instance Classification	Assigns individuals (ABox) to classes based on asserted and inferred property values and class definitions[3][6].
Consistency Checking	Detects logical contradictions within the ontology, such as violations of cardinality (only 0 or 1 allowed in OWL Lite), property constraints, or incompatible class assertions[1][6].
Property Characteristic Inference	Supports inference with transitive, symmetric, functional, and inverse properties. For example, infers new relationships via property characteristics (e.g., if P is transitive and aPb, bPc then aPc)[5][6].
Domain and Range Inference	Infers class membership of individuals based on property domain and range declarations (e.g., if aPb and P has domain C, then a is inferred to be an instance of C)[5].

3.2. LUBM Ontology

The Lehigh University Benchmark (LUBM) is a widely used benchmark for evaluating the performance of Semantic Web knowledge base systems, particularly those supporting OWL reasoning[14]. The LUBM ontology is designed to model a university domain, providing a structured framework for representing entities such as students, faculty, organizations, and academic programs.

The ontology defines a comprehensive set of classes and relationships relevant to university life. Notably, it includes 43 classes and 32 properties, with 25 object properties (relationships between classes) and 7 datatype properties (attributes with literal values). Key classes include `Person`, `Student`, `Employee`, `Dean`, `TeachingAssistant`, `Organization`, `Program`, `University`, `Work`, `Course`, `Unit`, `Stream`, and `Graduate Course`. Relationships such as `author`, `member`, `degreeFrom`, `masterDegreeFrom`, and `takesCourse` connect these classes, reflecting real-world interactions in an academic environment[15].

A distinctive feature of the LUBM ontology is its use of OWL Lite language constructs, including inverse properties (`inverseOf`), transitive properties (`Transitive Property`), some-value restrictions (`someValuesFrom`), and intersections (`intersecti onOf`). This allows for a moderate level of expressivity while maintaining computational tractability, making the ontology well-suited for benchmarking systems with varying reasoning capabilities. The LUBM benchmark is accompanied by scalable synthetic datasets that represent universities and their constituents, enabling controlled experiments and repeatable performance evaluations. The benchmark also includes a set of 14 extensional queries that test a variety of reasoning and retrieval tasks, ranging from simple class membership checks to complex relationship traversals. In summary, the LUBM ontology serves as a robust, standardized test bed for assessing the scalability, reasoning, and querying capabilities of Semantic Web systems, with a particular focus on OWL-based knowledge bases.

3.3. Making LLM Awareness of LUBM Ontology

The mechanism by which the LUBM ontology was provided to the LLM is mainly based on RAG (Retrieval-Augmented Generation) [16]. RAG is an advanced technique that enhances LLMs by dynamically integrating external, up-to-date information into their responses. Unlike traditional LLMs, which rely solely on static training data, RAG first retrieves relevant documents or data from external sources (e.g., databases, document repositories, or web pages) based on the user's query. The retrieved information is then combined with the original prompt and fed into the LLM, enabling it to generate more accurate, contextually relevant, and factually grounded answers [17]. This approach not only improves the model's performance and reduces the risk of hallucinations (incorrect or fabricated information), but also allows LLMs to access domain-specific or proprietary knowledge without the need for costly retraining or model updates [18]. In particular, Fig. 1 shows the workflow by means, in the proposed approach, the LLM answers the user's query layering on the LUBM ontology. More in detail, the document provided in RAG mode is the LUBM ontology processed through text embedding and stored into a Vector DB, namely PGVector[2]. Therefore, when the query arrives at the system, it is vectorized through text embedding and used to search the knowledge base (LUBM ontology). The search result is then attached, as context, to the original prompt (with the user's question) and sent to the LLM, which in turn answers the question.

3.4. Evaluation Approach

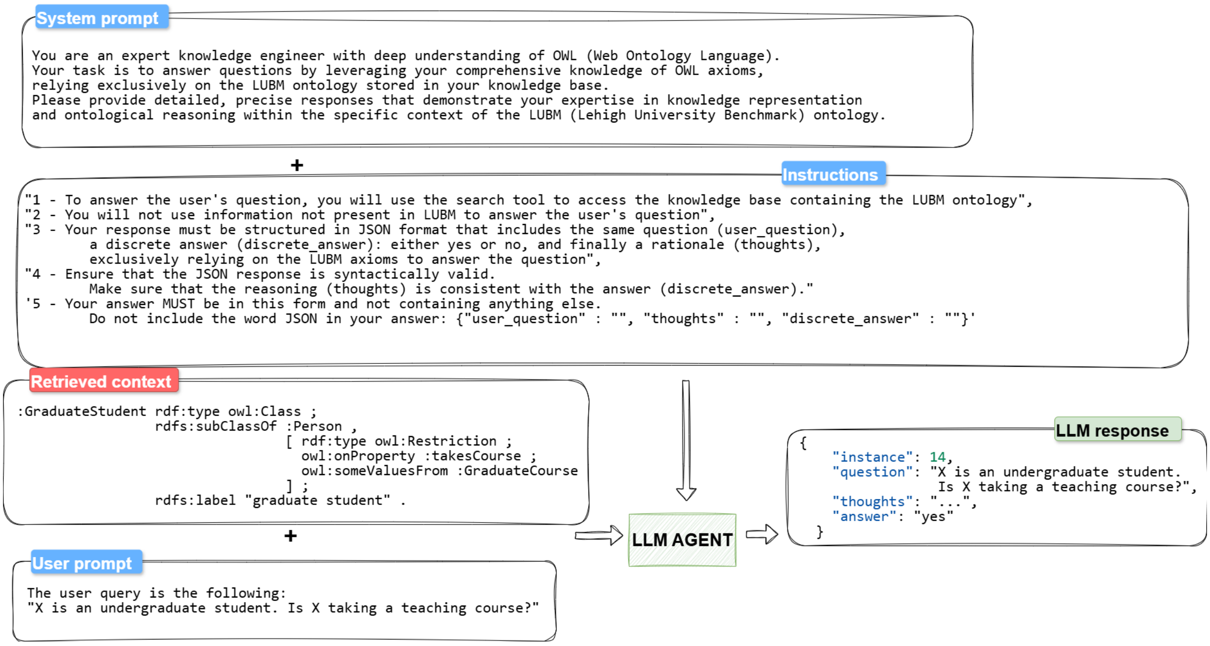
In order to assess and compare the ability of LLMs to read, understand, and make inferences on OWL Lite ontologies, a comprehensive and manually annotated set of challenging queries was designed. Such a set is made up of 53 queries in natural language, emulating the definition of individuals and reasoning over the LUBM ontology. Tab. 1 outlines the different facets of reasoning under assessment. Categorizing the queries in this way enables a more comprehensive evaluation of results, helping to identify model flaws and biases based on the types of inference involved. An additional layer of categorization is provided by the *level* of each query, which ranges from 1 to 3 and reflects the complexity of the logical reasoning required to answer it. *level 1* includes queries that involve direct reasoning, requiring only a single inference step. *level 2* comprises queries that necessitate two concatenated inference operations, representing moderately complex reasoning. Finally, *level 3* encompasses queries that demand advanced reasoning, involving three concatenated inference steps. The distribution of queries across these levels in the evaluation set is as follows: *level 1* – 55%, *level 2* – 38%, and *level 3* – 7%. The following element portrays an example of two items utilized in this evaluation:

```
[   {
        "id": 13,
        "premise": "X is an undergraduate student.",
        "query": "Is X taking a teaching course?",
        "correct answer": "Yes",
        "level": 2,
        "type": "instance-class equivalence"
    },
    {
        "premise": "X is part of Y. X is part of Z.",
        "query": "Is Y part of Z?",
        "correct_answer": "No",
        "level": 1,
        "id": 29,
        "type": "transitive property"
    },....]
```

Each evaluation item is composed of a *premise*, i.e., a statement that encodes an assertion about an individual within the ontology, serving as the foundation for the *query*. As stated in the following sections, these two fields can be considered together, forming a *full_query* field to pass as input to the model for the evaluation. To ensure a robust evaluation, a diverse set of LLMs was utilized, varying both in size and training methodology, allowing analyzing how these factors affect the ability to develop different reasoning over an ontology. In this evaluation setup, unlike classical OWL reasoners, the models were instructed to operate under a Closed World Assumption, meaning that any information not explicitly stated in the ontology is considered false. Future work could extend the evaluation to the

---

[2] https://github.com/pgvector/pgvector



```
You are an expert knowledge engineer with deep understanding of OWL (Web Ontology Language).
Your task is to answer questions by leveraging your comprehensive knowledge of OWL axioms,
relying exclusively on the LUBM ontology stored in your knowledge base.
Please provide detailed, precise responses that demonstrate your expertise in knowledge representation
and ontological reasoning within the specific context of the LUBM (Lehigh University Benchmark) ontology.
```

```
"1 - To answer the user's question, you will use the search tool to access the knowledge base containing the LUBM ontology",
"2 - You will not use information not present in LUBM to answer the user's question",
"3 - Your response must be structured in JSON format that includes the same question (user_question),
      a discrete answer (discrete_answer): either yes or no, and finally a rationale (thoughts),
      exclusively relying on the LUBM axioms to answer the question",
"4 - Ensure that the JSON response is syntactically valid.
      Make sure that the reasoning (thoughts) is consistent with the answer (discrete_answer)."
'5 - Your answer MUST be in this form and not containing anything else.
      Do not include the word JSON in your answer: {"user_question" : "", "thoughts" : "", "discrete_answer" : ""}'
```

```
:GraduateStudent rdf:type owl:Class ;
                 rdfs:subClassOf :Person ,
                                 [ rdf:type owl:Restriction ;
                                   owl:onProperty :takesCourse ;
                                   owl:someValuesFrom :GraduateCourse
                                 ] ;
                 rdfs:label "graduate student" .
```

```
The user query is the following:
"X is an undergraduate student. Is X taking a teaching course?"
```

```
{
    "instance": 14,
    "question": "X is an undergraduate student.
                Is X taking a teaching course?",
    "thoughts": "...",
    "answer": "yes"
  }
```

Figure 2: Prompt setting utilized for obtaining models' responses

Open World Assumption (OWA) to assess the models' ability to distinguish whether a query can be definitively answered with the available information. The models were configured to return a structured response consisting in their final answer to the query (only replying with "yes" or "no") and their reasoning process. The models' coherence was then evaluated, i.e., whether the reasoning process was consistent with the final answer.

3.5. Prompting

An important part of obtaining reliable results from the LLMs consists of providing an input prompt clearly explaining the task. The prompt was designed following common principles of prompt engineering, such as clarity, specificity, chain of thoughts, and role assignment [19]. More specifically, the defined prompt can be divided into system prompt, instructions, and user's prompt. The system prompt, used to recall the general knowledge of the models about OWL. Moreover, the instructions guide models to read the LUBM ontology, retrieved from the knowledge base (provided in RAG modality), and apply their knowledge about OWL on such ontology to answer the user's questions. The instructions try to avoid pre-training knowledge that possibly conflicts with the provided LUBM ontology. This also ensures, to a certain extent, a correct evaluation of reading, comprehension and inference of models. Lastly, instructions put also importance on aligning the reasoning process of the model with their response, and on the structured generation of the response as a JSON object as depicted in Fig. 2.

4. Experimentation and Evaluation

4.1. Settings

As it was anticipated in section 3.4, the results were compared on a diverse set of LLMs for exploring the differences in the results according to model size, architecture, or training process. Only open-source or open-weights models were considered in the set. This includes different version of Llama models, spanning from smaller (`Llama 3.1 8b`)[3] to bigger size in parameters and newer versions (`Llama`

---

[3] https://huggingface.co/meta-llama/Meta-Llama-3-8B

Table 2
Summary of the LLMs utilized in the experiment

Model	N. Active Parameters	Key Features
Llama 3.1 8b	8b	Smaller model from the Llama 3 family; used to assess performance at reduced parameter scale.
Llama 3 70b	70b	Large-scale version of Llama 3, included for evaluating performance improvements with more parameters.
Llama 3.3 70b versatile	70b	Newer version of Llama 3.3, aimed at capturing more recent training improvements.
Llama4-Maverick	17b - 128 experts	Based on a Mixture of Experts (MoE) architecture; used to explore effects of expert routing on inference.
qwen-qwq-32b	32b	Mid-sized model trained with Reinforcement Learning (RLHF); included to test models incorporating feedback-based optimization.
Deepseek-R1 distill-llama-70b	70b	Distilled version of Deepseek-R1 using Llama-based architecture;intended to evaluate performance of a distilled model.

`3 70b`[4], `Llama 3.3 70b versatile`[5]). Also, the `Llama4-Maverick`[6] model was included for evaluating the performance of a Mixture of Experts architecture. The `qwen-qwq-32b`[7] model was added for evaluating a mid-size model incorporating Reinforcement Learning in its training process. Lastly, we also included a model based on the Llama architecture, trained via distillation using the `Deepseek-R1` model[8]. Tab. 2 outlines the main features of the selected models.

The phi-data framework[9] was utilized for creating agents with well defined roles and contextual knowledge. The OWL Lite ontology in textual format was chunked and embedded into vectors utilizing the PGvector[10] PostgreSQL extension for performing similarity searches. The agents were then connected to the LLMs through the Groq[11] API and Ollama[12] for running local models. All experiments were conducted on a machine equipped with 16 GB of RAM and an NVIDIA GeForce RTX 4050 Laptop GPU with 6 GB of dedicated VRAM.

4.2. Experiments

For each model in Tab. 2, an agent was created by incorporating the system prompt and instructions as described in Section 3.5. Each agent was instructed to skip the internal step-by-step reasoning (via the `reasoning` parameter), making them equivalent to a plain LLM invocation with RAG, where the context is simply retrieved and appended to the prompt. The manually annotated item in the set of queries was passed to the agent to obtain the structured response, which consisted of a discrete answer (yes or no) and a rationale (referred to as "thoughts") explaining the reasoning behind the answer. To ensure robustness in the evaluation process, a recursive function was employed for each query. This function attempted to generate a valid structured response, and in the event of a failure (e.g., malformed output or parsing errors), it automatically retried the generation process up to five times. This was necessary due to occasional inconsistencies in the model's output format. Eventually, the metric used to evaluate performance was classification accuracy, computed by comparing the discrete answers to the ground-truth annotations.

---

[4] meta-llama/Llama-3.1-70B

[5] https://console.groq.com/docs/model/llama-3.3-70b-versatile

[6] https://console.groq.com/docs/model/llama-4-maverick-17b-128e-instruct

[7] https://huggingface.co/Qwen/QwQ-32B

[8] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B

[9] https://www.phidata.app/

[10] https://github.com/pgvector/pgvector

[11] https://console.groq.com/home

[12] https://ollama.com/

4.3. Results

Table 3
Model's Accuracy on the set of 53 queries, detailed according to the level of complexity of the query.

model	**accuracy**	**level1 accuracy**	**level2 accuracy**	**level3 accuracy**
llama3-70b	79%	72%	85%	**100%**
llama3.3-70b-versatile	85%	90%	75%	**100%**
llama4-maverik 17b-128e-instruct	**94%**	**93%**	**95%**	**100%**
deepseek-r1-distill-llama-70b	85%	86%	85%	75%
qwen-qwq-32b	77%	73%	80%	**100%**
llama3.1-8b	64%	55%	75%	50%

Tab. 3 states the accuracy results obtained from the models on the complete set and over the three complexity levels. Llama 4 obtains a remarkable 94% overall accuracy and the highest scores for each of the three levels, suggesting the ability of a Mixture of Expert architecture [20] in such a task. The results also indicate lower performance obtained by smaller size models-`qwen-qwq-32b` and `llama3.1-8b`- as well as `llama3-70b`, i.e., a model not having updated training and architecture. The difference in performance compared to the other models is particularly evident already in *level 1* queries, suggesting that even queries requiring only a few logical steps are challenging for them.

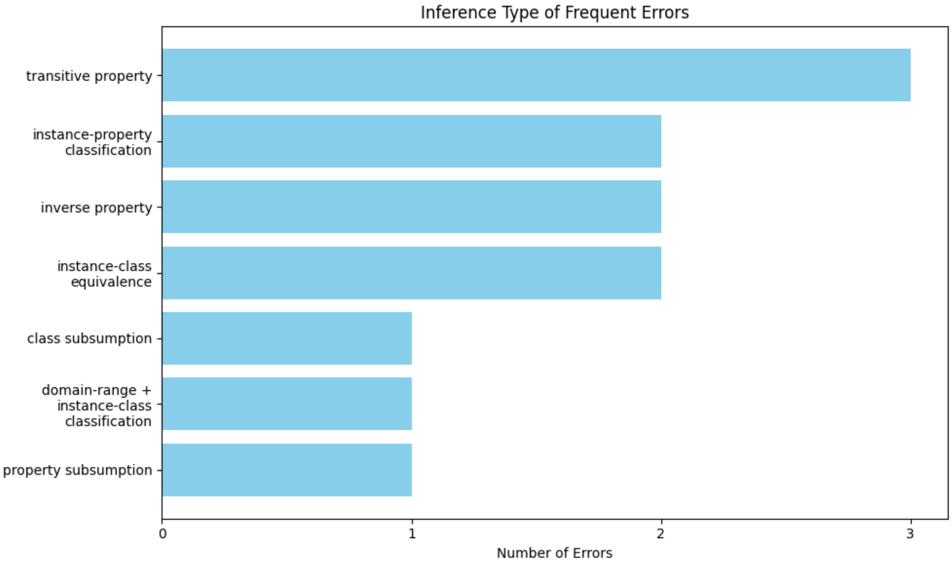


Figure 3: Inference type of queries misclassified by at least two models.

Moreover, Fig 3 reports common errors related to the same inference types, i.e., queries incorrectly answered by at least two models. Queries based on the transitive property lead to the highest number of errors, suggesting models' difficulties in applying transitive reasoning. Fig 4 reports the level of complexity of the commonly misclassified queries, stressing that the models' classification for these inference types was problematic even for *level 1*. The figure reports that the 24.5% of *level 1* and 25% of *level 2* queries were wrongly classified by more than one model. This even distribution indicates that the errors depended more on the inference type than on the level of complexity. Also, it is notable that there are no common *level 3* errors, probably mainly due to their low representation in the set considered.

Fig. 5 presents an analysis of the errors model by model, with a focus on the underlying inference types. While the models exhibit different patterns of error—indicating varied strategies for addressing

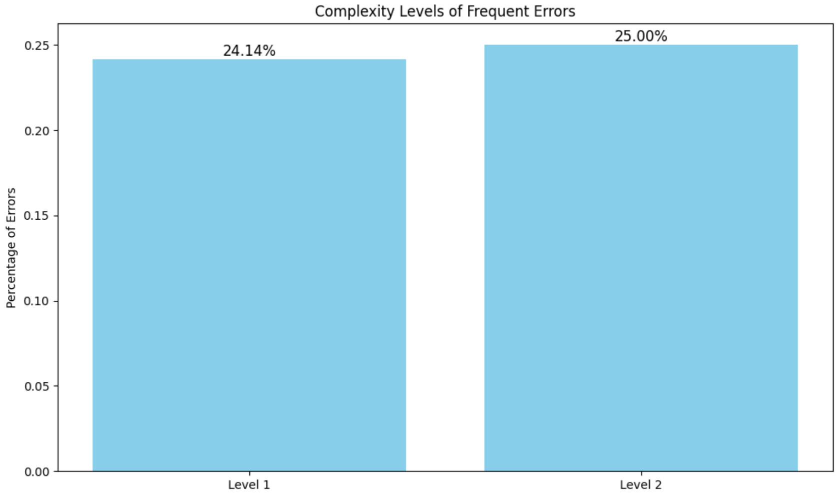


Figure 4: Levels of complexity of queries misclassified by at least two models.

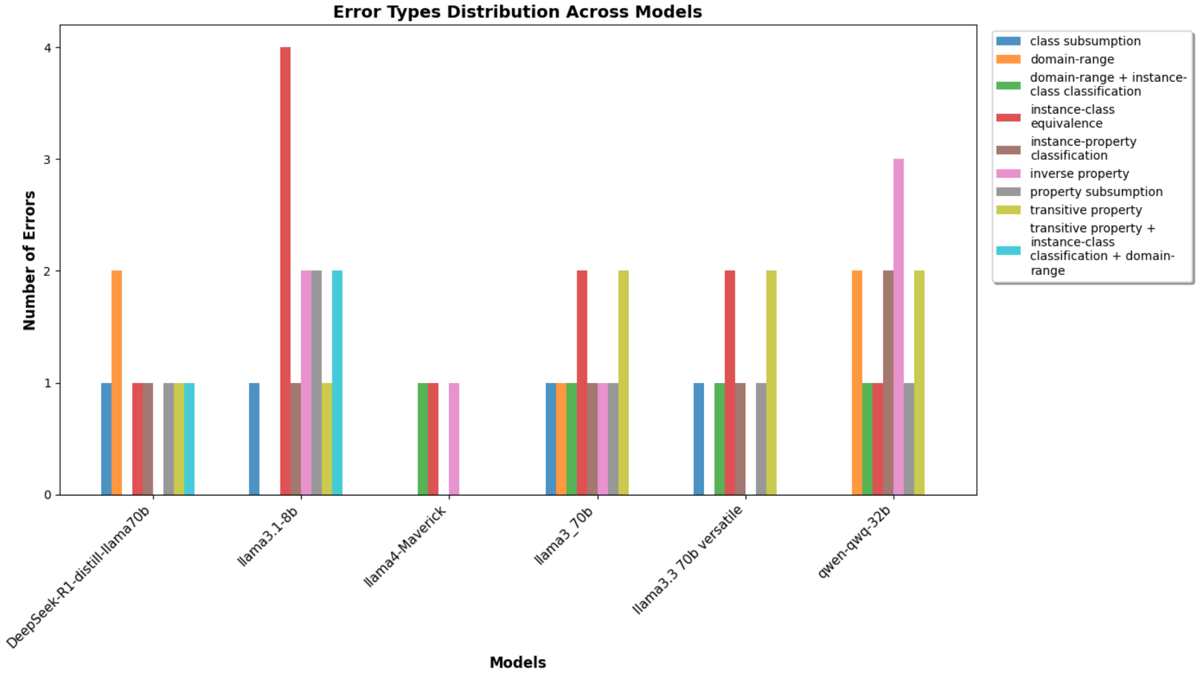


Figure 5: Number of errors obtained from each LLM conditioned by inference type

the queries—some commonalities emerge that warrant further examination.

As anticipated by Fig. 3, the models' struggles with 'transitive-property', since it was the most present inference type among the frequent errors, and 5 out of 6 models report at least one error of this kind. A closer look to the models' explanations for a query of level 1 of such a type provides details about the models' biases in the logical pattern. The considered query is analyzed as an example: *"X is part of Y. X is part of Z. Is Y part of Z?"*. The `qwen-qwq-32b` and `llama3-70b` wrongly answered with 'yes' and presented the common patterns in their explanations. Both the models misunderstood the logic of the transitive property, incorrectly assuming that *if X is part of Y and X is part of Z, we can conclude that Y is part of Z as well.* This logical flaw is repeated over queries of this type, showing an over extension of the reasoning, as the models are not relying just on axioms of the provided ontology, but they apply common associative reasoning. The `Llama4-Maverick` model is the one providing the best explanations, correctly recognizing the logic of transitive property and its caveats: *if X is part of Y* *and Y is part of Z, then X is part of Z. However, the question is asking whether Y is part of Z, given that X is part of Y and X is part of Z. The transitivity of 'subOrganizationOf' does not directly answer this question.*

Table 4
Model responses and reasoning for checking subproperty relations

model	explanation	discrete answer
llama3-70b	To determine if 'Is the Head Of' is a subproperty of 'Member Of', I need to check the subproperty axioms in the LUBM ontology. Specifically, I'll look for a subPropertyOf axiom that relates 'Is the Head Of' to 'Member Of'.	no
llama3.3-70b-versatile	To answer this question, we must first identify the properties 'Member Of' and 'Is the Head Of' in the LUBM ontology. Then, we should determine if 'Is the Head Of' is defined as a subproperty of 'Member Of'. This can be done by checking the property hierarchies defined in the LUBM ontology. If 'Is the Head Of' is indeed a subproperty of 'Member Of', it means that any instance related by 'Is the Head Of' is also related by 'Member Of'. This relationship is crucial for ontological reasoning and inference.	no
llama4-maverik 17b-128e-instruct	**The LUBM ontology defines the 'headOf' property as a subproperty of 'worksFor', and 'worksFor' is a subproperty of 'memberOf'. Therefore, 'headOf' is indirectly a subproperty of 'memberOf'.**	**yes**
deepseek-r1-distill-llama-70b	To determine if 'Is the Head Of' is a subproperty of 'Member Of', I need to check if there is a subproperty axiom in the LUBM ontology that defines 'Is the Head Of' as a subproperty of 'Member Of'.	no
qwen-qwq-32b	To answer this question, I need to check if 'Is the Head Of' is a subproperty of 'Member Of' in the LUBM ontology. This can be done by searching for the rdfs:subPropertyOf axiom that relates 'Is the Head Of' to 'Member Of'.	no
llama3.1-8b	To determine if 'Is the Head Of' is a subproperty of 'Member Of', we need to analyze their definitions and relationships within the LUBM ontology.	no

A similar pattern applies for the 'instance-class equivalence which appears at least once as an error in all the models considered. The error analysis is enlarged by examining one of the trickiest queries in the set, being correctly classified only by the Llama4-Maverick model: *'Member Of' is a property Is 'Is the Head Of' a subproperty of 'Member Of'?*

Models' responses are reported in Tab. 4. In particular, Llama4-Maverick identifies the correct properties and is able to reason considering its sub-properties. For the other models, it is notable that although the reasoning appears generally correct, it is limited to the task description. This also appears too vague in the case of llama3.1-8b and too verbose in the case of llama3.3-70b-versatile. This pattern emerges consistently across misclassified queries, regardless of inference type and level of complexity, depicting a lack of coherence between the rationale and the final answer. This is especially true for the smallest model -`llama3.1-8b`- while it appears with less frequency in the newest version of llama -`llama4-Maverick`- indicating greater logical understanding and coherence from a model with an updated training and a Mixture of Experts architecture.

5. Discussion and Future Works

This work investigated how LLMs can read, understand, and perform logical inferences over an OWL Lite ontology. The well-known LUBM ontology was employed as a reference symbolic model due to its coverage of diverse inference types, including inverse properties, transitive properties, some-value restrictions, and class intersections. The LLMs' knowledge was constrained to only the supplied LUBM ontology, excluding prior notions from their training data. Their inference ability was then evaluated through a dedicated set of manually annotated binary queries, allowing the assessment of their capacity to understand and generalize the concepts expressed in the ontology and apply them to new contexts and individuals. The proposed approach consists of two phases. The first phase makes the OWL Lite ontology accessible to the LLM by transforming it into a dense vector representation. The evaluation query is also transformed, and using similarity scores, the relevant context is retrieved and assembled into a prompt for obtaining both the discrete answer and self-explanation from the LLM. The second phase involves extensive results evaluation, considering the inference types supported by OWL Lite. The results evaluation in Section 4.3 identified `Llama4-Maverick` as the best one in terms of accuracy and quality of reasoning, evaluated through the models' self-explanations. The results also suggested that models are more affected by the type of reasoning than by query complexity. In particular, 'transitive-property', was identified as the most problematic inference type. Overall, this work provides insights into LLM reasoning when grounded in symbolic ontological knowledge, thereby contributing to the development of semantically aware AI systems. Demonstrating that LLMs can perform reasoning following rules and constraints expressed by an ontological schema underscores their potential to generate grounded, explainable, and logically consistent responses. This aligns with the broader vision of Generative eXplainable AI (GenXAI), moving beyond purely data-driven outputs toward AI systems that are both interpretable and anchored in structured domain knowledge. Considering this work as foundational, several future research directions are identifiable. The evaluated query set and the LLMs input instructions are a fundamental part of the approach as they shape the models' behavior and the basis for the evaluation. Two important assumptions were made in this work to reduce the overall complexity. The queries' answers were limited to yes/no; future works may explore the behavior of the models in a multiclass setting. Similarly, the Open World Assumption (OWA) was not considered. For a comprehensive evaluation of LLMs as logical reasoners, it is important to assess their ability to distinguish between queries that can be definitively answered and those that cannot be determined under OWA.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI GPT-4 for grammar and spelling check. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] G. Lei, R. Docherty, S. J. Cooper, Materials science in the era of large language models: a perspective, Digital Discovery 3 (2024) 1257–1272. URL: https://doi.org/10.1039/d4dd00074a. doi:10.1039/d4dd00074a.

[2] J. Chu-Carroll, A. Beck, G. Burnham, D. Melville, D. Nachman, A. E. Özcan, D. Ferrucci, Beyond llms: Advancing the landscape of complex reasoning, arXiv (Cornell University) (2024). URL: http://arxiv.org/abs/2402.08064. doi:10.48550/arxiv.2402.08064.

[3] L. Pan, A. Albalak, X. Wang, W. Y. Wang, Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning, arXiv (Cornell University) (2023). URL: https://arxiv.org/abs/2305.12295. doi:10.48550/arxiv.2305.12295.

[4] G. Antoniou, F. v. Harmelen, Web ontology language: Owl, Handbook on ontologies (2009) 91–110.

[5] J. Beverley, F. Franda, H. Karray, D. Maxwell, C. Benson, B. Smith, Ontologies, arguments, and large-language models (2024).

[6] W. Gibaut, L. Pereira, F. Grassiotto, A. Osorio, E. Gadioli, A. Munõz, S. Gomes, C. dos Santos, Neurosymbolic ai and its taxonomy: a survey, arXiv (Cornell University) (2023). URL: https://arxiv.org/abs/2305.08876. doi:10.48550/arXiv.2305.08876.

[7] M. Gaur, A. Sheth, Building trustworthy neurosymbolic ai systems: Consistency, reliability, explainability, and safety, AI Magazine 45 (2024) 139–155. URL: https://doi.org/10.1002/aaai.12149. doi:10.1002/aaai.12149.

[8] J. Schneider, Explainable generative ai (genxai): A survey, conceptualization, and research agenda, Artificial Intelligence Review 57 (2024) 289.

[9] P. Mondorf, B. Plank, Beyond accuracy: Evaluating the reasoning behavior of large language models – a survey, arXiv (Cornell University) (2024). URL: http://arxiv.org/abs/2404.01869. doi:10.48550/arxiv.2404.01869.

[10] C. Agarwal, S. H. Tanneru, H. Lakkaraju, Faithfulness vs. plausibility: On the (un)reliability of explanations from large language models, arXiv (Cornell University) (2024). URL: https://arxiv.org/abs/2402.04614. doi:10.48550/arXiv.2402.04614.

[11] Y. He, J. Chen, E. Jiménez-Ruiz, H. Dong, I. Horrocks, Language model analysis for ontology subsumption inference, 2023. URL: https://arxiv.org/abs/2302.06761. arXiv:2302.06761.

[12] K. Wang, G. Qi, J. Li, S. Zhai, Can large language models understand dl-lite ontologies? an empirical study, 2024. URL: https://arxiv.org/abs/2406.17532. arXiv:2406.17532.

[13] D. Beckett, T. Berners-Lee, E. Prud'hommeaux, G. Carothers, Rdf 1.1 turtle, World Wide Web Consortium (2014) 18–31.

[14] Y. Guo, Z. Pan, J. Heflin, Lubm: A benchmark for owl knowledge base systems, Web Semant. 3 (2005) 158–182. URL: https://doi.org/10.1016/j.websem.2005.06.005. doi:10.1016/j.websem.2005.06.005.

[15] S. Jean, L. Bellatreche, G. Fokou, M. Baron, S. Khouri, Ontodbench: Novel benchmarking system for ontology-based databases, in: R. Meersman, H. Panetto, T. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, I. F. Cruz (Eds.), On the Move to Meaningful Internet Systems: OTM 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 897–914.

[16] Y. Zhou, Y. Liu, X. Li, J. Jin, H. Qian, Z. Liu, C. Li, Z. Dou, T.-Y. Ho, P. Yu, Trustworthiness in retrieval-augmented generation systems: A survey, 2024. doi:10.48550/arXiv.2409.10102.

[17] Y. Huang, J. X. Huang, A survey on retrieval-augmented text generation for large language models, ArXiv abs/2404.10981 (2024). URL: https://api.semanticscholar.org/CorpusID:269188036.

[18] W. Zhang, J. Zhang, Hallucination mitigation for retrieval-augmented large language models: A review, Mathematics 13 (2025). URL: https://www.mdpi.com/2227-7390/13/5/856. doi:`10.3390/math13050856`.

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

[20] A. Vats, R. Raja, V. Jain, A. Chadha, The evolution of mixture of experts: A survey from basics to breakthroughs, Preprints (August 2024) (2024).