Ontology Reuse in LLM-Generated Ontologies: A Comparative Evaluation Framework

Henon Mengistu Lamboro[1,2,*], Amel Bouzeghoub[1], Cécile Rabrait[2], Antonio Kung[2], Olivier Genest[2] and Amelie Gyrard[2]

[1]SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau, France
[2]Trialog, 75008 Paris, France

Abstract

Ontologies are meant to be reused; however, reusing ontologies in practice is highly challenging. Recent advances in Large Language Models (LLMs) offer new possibilities for ontology engineering and competency question generation, but their potential for explicit ontology reuse is underexplored. In this paper, we introduce a framework for evaluating ontology reuse in ontologies generated by LLMs. The framework measures reuse across four dimensions: lexical reuse, structural reuse, logical consistency, and reuse depth. We systematically examine ontology reuse in LLM-generated ontologies across four models (GPT-4o-mini, GPT-4.1, Qwen2.5-7B, and Llama-3.1-8B) with various prompting strategies using established energy and IoT domain ontologies such as SAREF and the Fiesta-IoT Ontology. We observe that unstructured prompts result in minimal or no reuse, while reuse-orientated prompting increases the number of classes aligned with existing ontologies. Our results demonstrate that while LLMs achieve lexical reuse with reference ontologies in a controlled reuse-oriented prompting technique, deep structural and axiomatic reuse remains limited. This study highlights that effective ontology reuse with LLMs requires dedicated prompting, alignment mechanisms, LLM-ready ontology reuse methodologies and hybrid human-in-the-loop workflows to ensure ontology reuse operations are considered and implemented.

Keywords

Ontology Reuse, LLM Evaluation, Prompt Engineering, Ontology Engineering

1. Introduction

Ontology reuse is a systematic adoption of previously developed ontologies or ontology components for creating new ontologies [1]. It is motivated by the need to reduce development costs, improve interoperability, ensure consistency across knowledge-based systems and accelerate the engineering lifecycle. Reusing ontologies has several advantages. Reusing established ontology components avoids modelling from scratch and leverages artefacts that have already been tested and validated [2]. In practice, ontology reuse relies on manual inspection, alignment, and conceptual understanding, processes that demand considerable expertise and cognitive effort.

Ontology repositories such as the OBO Foundry [3], BioPortal[4], and Linked Open Vocabularies (LOV) [5], along with tools such as Protégé [6], have facilitated the discovery and adoption of ontologies across multiple domains. These repositories provide access to hundreds to over a thousand distinct ontologies covering diverse subject areas: for example, BioPortal alone currently hosts over 1,500 ontologies, and LOV currently hosts 905 ontologies. These repositories and tools take on the effort of human experts in constructing models of domain knowledge and making reuse feasible. Nevertheless, one of the significant challenges remains: how to ensure meaningful, semantically consistent reuse in human or automated ontology engineering pipelines. The costs associated with the different steps

ELMKE 2026: Evaluation of Language Models in Knowledge Engineering, 3rd Workshop co-located with ESWC 2026, May 10-11, 2026, Dubrovnik, Croatia
*Corresponding author.
henon.lamboro@ip-paris.fr (H. M. Lamboro); amel.bouzeghoub@telecom-sudparis.eu (A. Bouzeghoub); cecile.rabrait@trialog.com (C. Rabrait); antonio.kung@trialog.com (A. Kung); olivier.genest@trialog.com (O. Genest); amelie.gyrard@trialog.com (A. Gyrard)
0009-0001-5813-4584 (H. M. Lamboro); 0000-0003-4890-9005 (A. Bouzeghoub); 0000-0001-9384-8251 (C. Rabrait); 0000-0003-0042-6776 (A. Kung); 0009-0002-7305-0001 (O. Genest); 0000-0003-3938-5617 (A. Gyrard)

involved in reusing existing ontologies are not well understood or predetermined. On the other side, Simperl's feasibility study [7] on ontology reuse highlighted that, although numerous ontologies are available, developers face challenges such as heterogeneity, low discoverability, and limited incentives, which hinder systematic reuse. This work reframed ontology reuse as a socio-technical challenge rather than a purely technical one. Many research groups and organisations develop their methodologies to reuse existing ontologies and to develop new ontologies tailored to their needs. Varying and inconsistent approaches taken by ontology experts hinder the understanding and application of ontology reuse. Choosing the right ontologies and knowing how to reuse them is an important basis for successfully implementing ontologies and carrying out knowledge graph engineering tasks while achieving semantic interoperability.

LLMs have increasingly become a primary area of interest in research due to their potential in various natural language processing tasks, including summarisation, text generation, code generation, and question answering [8, 9, 10]. More recently, the semantic web research community has turned attention to how LLMs can support semantic technologies [11, 12]. LLMs can generate ontologies, OWL axioms, class hierarchies, and RDF triples from natural language inputs. However, these automated approaches often operate without explicit awareness of existing ontologies, potentially leading to semantic inconsistencies. As Garijo et al. [11] highlight in their systematic landscape review, the **reuse phase remains underdeveloped**, as no LLM-based approaches currently support explicit ontology search, selection, or adaptation. In addition, the **evaluation phase lacks systematic frameworks** for assessing the level and quality of ontology reuse in LLM-generated ontologies. While some efforts have focused on evaluating ontology generation, for example through benchmarks such as the one proposed by Plu et al. [12], which assess LLM-generated ontologies using metrics such as accuracy, precision, recall, coverage, and qualitative user evaluation, explicit ontology reuse by LLMs remains largely underexplored. These gaps highlight the need for more mature ontology reuse methodologies to effectively integrate LLMs into ontology engineering. These challenges become even more pronounced in emerging automated settings, particularly with the rise of LLM-based ontology generation.

This paper attempts to address these gaps by conducting a comparative evaluation of ontology reuse in LLM-generated ontologies. Understanding the maturity, scope, and limitations of LLMs regarding ontology reuse is essential for advancing the field in a structured and reliable way. Through this comparison, the paper seeks to answer the following research question:

- **RQ:** Do LLM-generated ontologies meaningfully reuse existing ontological components?

This work has three main contributions. First, it introduces a **reproducible evaluation framework** for assessing ontology reuse in LLM-generated ontologies. Second, it presents the **first empirical comparison** of ontology reuse in various LLMs, covering smart energy and IoT domains. Third, it articulates a **research agenda** highlighting the need for explicit reuse-aware methodologies in LLM-based approaches.

The remainder of this paper is structured as follows. Section 2 reviews related work on ontology reuse and recent developments in LLM-based ontology engineering. Section 3 describes the framework for evaluating ontologies generated by LLMs. Section 4 details the experimental design and steps, including prompting techniques, metrics, and the evaluation workflow. Section 5 presents the results and analysis. Section 6 concludes and outlines research directions. All results and documents are available on GitHub: https://github.com/HenonLamboro-Trialog/OntologyReuse-LLM-Evaluation.

2. Background

2.1. Ontology

There are many definitions of the term "ontology" used in the semantic web. The one from Gruber is the most relevant and most cited. In accordance with Tom Gruber's [13] definition, ontology is the "explicit specification of conceptualisation". Another definition by Struder [14] built on Gruber's definition and slightly elaborated it as "An ontology is a formal, explicit specification of a shared conceptualisation."

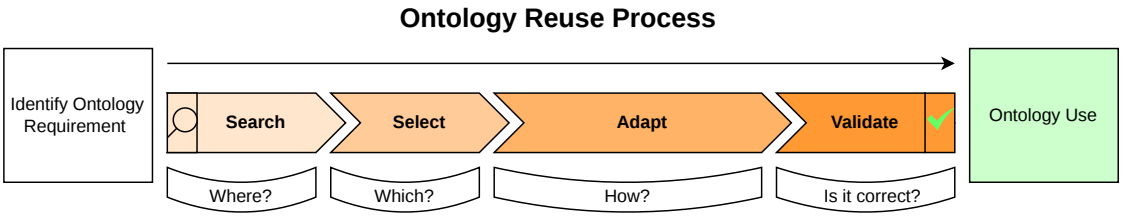

Figure 1: Ontology Reuse Process

Ontologies are a means to formally model the structure of a system in both human-understandable and machine-readable formats. They are used to formalise and keep the relationship of data in a system. Concepts, properties, and relations are explicitly defined to represent machine-readable abstract models.

Ontologies have been widely recognised as a core component in the development of the Semantic Web, serving as the foundation for enhancing semantic interoperability across diverse systems and data sources. Moreover, as outlined in the FAIR guiding principles[15], reusability is one of the attributes for maximising the added value of information artifacts.

2.2. Ontology Reuse

The term 'ontology reuse' has been around since the late 1990s, and it has been studied for years. One of the earliest pieces of literature that mentioned ontology reuse is the paper entitled "An Experiment in Ontology Reuse" by Uschold et al. [16], where ontology reuse is investigated as a key element in ontology engineering for a specific domain software design. Afterwards, a more formal and foundational definition is presented by Katsumi and Grüninger [17]. They formalise ontology reuse for the first time as a mathematical problem by giving more than fifteen mathematical definitions of reuse operations. They also critically examined the notion of reuse and categorised different ontology reuse strategies. Building on these foundational studies, ontology reuse has come to be widely recognised as a fundamental principle of ontology engineering and a key driver of the Semantic Web vision. Reuse enables developers to build upon existing conceptualisations, thereby promoting semantic interoperability, accelerating development time, and improving model quality, as highlighted in [18]. As emphasised in earlier studies, ontology reuse contributes to faster development cycles by reducing engineering time by leveraging existing work, improving ontology quality, better aligning heterogeneous systems and encouraging the adoption of widely accepted ontologies. As such, ontology reuse is not only helpful but essential in ontology engineering.

In this paper, ontology reuse is defined as the process of accessing, selecting, and applying appropriate operations to existing ontologies in order to construct a new ontology that satisfies specified requirements as illustrated in Figure 1. Ontology use is the ability to utilise ontologies without modification for future applications, such as knowledge graph engineering, inference and reasoning. To make ontology reuse more concrete, consider an example from the RESONANCE ontology [19], where SRI4ALL aligns with SAREF4ENER [20], reusing energy-related concepts (e.g., `sri4all:Device rdfs:subClassOf s4ener:Device`).

2.2.1. Formalization of Ontology reuse

Here, we present formal definitions of ontology reuse and reusability, as introduced by [17]. Their work aimed at providing mathematical precision to the concepts of reuse, reusability, and the operations used to manipulate ontologies in the reuse process.

An ontology T is **reusable** for a set of intended models M_{intended} if there exists a subtheory $T' \subseteq T$ and signatures $\sigma_1 \subseteq \sigma(M_{\text{intended}})$ and $\sigma_2 \subseteq \sigma(T')$ such that:

$$\exists \pi : \text{Red}(M_{\text{intended}}, \sigma_1) \rightarrow \text{Red}(\text{Mod}(T'), \sigma_2) \tag{1}$$

The reduct of a class of models M to a signature σ is defined as:

$$\text{Red}(M, \sigma) = \{N : N \cong M|_\sigma,\ M \in M\} \tag{2}$$

Mod(T) is the models of an ontology T.

A key aspect of such formalisation of ontology reuse is categorising the ways an existing ontology can be manipulated to support a new modelling effort.

2.3. LLMs for Ontology Engineering

Recent advancements in LLMs have shown growing interest for their application to ontology engineering. LLMs have shown promising results in tasks such as ontology generation, concept extraction, axiom suggestion, and terminology alignment. For example, works like [21, 22, 23, 24, 25, 26, 27, 28] explored various prompting strategies and human-in-the-loop pipelines to build ontologies from text or structured sources. These efforts reflect a trend toward leveraging LLMs as general-purpose semantic agents capable of accelerating ontology engineering. Recent work further supports this by proposing methods that leverage LLMs to retrofit competency questions from existing ontologies, helping to recover implicit requirements and better understand their intended scope, which facilitates their effective reuse [29].

However, when it comes to ontology reuse, the landscape becomes less mature. Despite the growing sophistication of LLMs, their potential to identify, adapt, and integrate existing ontologies has received limited investigation. The reuse of established ontologies, such as foundational upper ontologies or domain-specific ontologies, involves tasks like ontology alignment, modularisation, context-aware reuse, and semantic compatibility, which are not trivially addressed by LLM prompting. LLMs often hallucinate structures or invent terminology, which may lead to ontologies that are semantically isolated and difficult to interoperate with existing knowledge graphs or systems.

3. Framework for Evaluating Ontology Reuse

This section presents the experimental evaluation of ontology reuse in LLM-generated ontologies using our proposed reuse framework. The evaluation follows the designed methodology described in Figure 2, where ontology reuse is operationalised across lexical, structural, semantic, and logical levels. The goal of the experiment is to analyse how the choice of different prompting strategies and LLMs influence ontology reuse when generating ontologies for the energy and Internet of Things (IoT) domain.

3.1. Experimental Workflow

As illustrated in Figure 2, the experiment consists of seven main stages:

1. **Prompt design:** A set of prompts was designed following three prompting strategies: *unstructured prompting*, *structured prompting*, and *reuse-oriented prompting*. Each prompt requested the generation of an ontology describing the smart appliance and IoT domain.
2. **Ontology generation:** Ontologies were generated using multiple LLMs, including GPT-4.1, GPT-4o-mini, Llama-3.1-8B-Instruct, and Qwen2.5-7B-Instruct. Each generated ontology was exported in Turtle (TTL) format and stored as a separate file.
3. **Preprocessing:** A preprocessing step was applied to ensure valid ontology syntax by removing markdown formatting, code block markers, and other non-RDF artefacts introduced by the LLMs.
4. **Ontology parsing:** The cleaned ontologies were parsed using the RDFLib library in Python [30], representing each ontology as an RDF graph of subject–predicate–object triples.
5. **Feature extraction:** Ontology features were automatically extracted from the parsed graphs, including lexical terms, hierarchical relations, property patterns, subsumption relations, and logical constraints.
6. **Reuse metrics computation:** Reuse metrics were computed by comparing extracted features with reference ontologies.

7. **Visualization and Analysis:** Results were exported for visualisation and analysis.

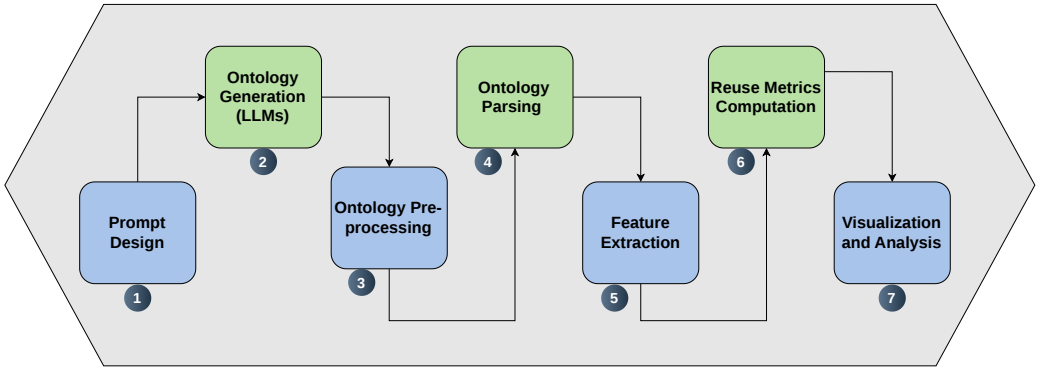

Figure 2: Framework for ontology generation and reuse evaluation.

4. Experimental Setup

To assess the proposed framework, we conducted experiments with 24 LLM-generated ontologies against reference ontologies with various prompting strategies. Batch evaluation was automated via a Python pipeline.

4.1. Prompting Strategies

Three prompting strategies with increasing levels of guidance were defined in order to test whether LLMs spontaneously reuse ontology terms without guidance.

1. **Unstructured Prompting**: A zero-shot baseline where the model is given only a high-level task description without structural or reuse guidance. For example: *"Generate an ontology for the smart appliance domain."*
2. **Structured Prompting**: The model is given explicit structural and output instructions without reuse constraints to test whether structured guidance improves alignment with domain terminology. *For example: Generate an ontology for smart appliances with at least 30 classes, 8 object properties, and 5 data properties, including domains and ranges.*
3. **Reuse-Oriented Prompting**: The model is explicitly encouraged to reuse terminology from an existing ontology, for example: *"Generate an ontology for the smart appliance domain by reusing Ontology X classes whenever appropriate. Prefer reused labels over new ones."*

For reproducibility, all prompts used in the experiments are provided in Appendix A and are available in the project repository. These prompts were applied consistently across all evaluated LLMs.

4.2. Ontology Generation Setup

Ontology generation was performed using four LLMs two larger proprietary GPT models and two open models, namely GPT-4o-mini [31], GPT-4.1 [32], Llama-3.1-8B-Instruct [33], and Qwen2.5-7B-Instruct [34], resulting in a total of 24 ontologies. All models were queried through a Python-based pipeline that automatically stores outputs in Turtle format. The temperature parameter was fixed at 0.2 for all models, reducing randomness while preserving limited variability in generation.

All prompts follow a shared ontology requirement to ensure comparability across outputs. For each prompting strategy, each LLM generates one ontology within the domain of smart appliances, including

devices, sensors, and energy-related concepts. The ontologies include core modelling elements such as classes (e.g., `Appliance`, `Sensor`, `Function`), object properties, and data properties (e.g., energy consumption), and are expressed in RDF/Turtle. They remain lightweight, focusing on taxonomy and basic relations without deep axiomatisation. In reuse-oriented settings, they additionally require alignment with reference ontologies through reuse of existing terms. This shared setup ensures that differences in outputs are attributable to prompting strategies rather than task ambiguity.

4.3. Ontology Preprocessing and Parsing

A preprocessing step was applied to the generated outputs to ensure valid ontology syntax. This step removed markdown formatting, code block markers, and other non-RDF artefacts introduced by the LLMs, ensuring that the resulting files could be correctly parsed as turtle ontologies. Each ontology, both reference and generated, is expressed in turtle format and loaded into a machine-readable RDF graph using `RDFLib`. Parsing the ontology graph allows systematic traversal of all triples in the form $(subject, predicate, object)$, enabling extraction of ontology components necessary for reuse evaluation.

4.4. Feature Extraction

From each ontology graph, the following four sets are extracted to capture different dimensions of reuse:

- **Lexical Terms (T)**: all class and property labels defined in domain-specific namespaces. *Example:* labels such as `TemperatureSensor` or `measuresProperty`.
- **Hierarchical Relations (H)**: all subclass relations between domain-specific classes. *Example:* `TemperatureSensor` $\sqsubseteq$ `Sensor`.
- **Property Patterns (P)**: object properties whose domain and range are restricted to domain-specific classes. *Example:* `measuresProperty (Sensor` $\rightarrow$ `Property)`.
- **Logical Constraints (C)**: OWL restrictions applied to classes. *Example:* `Sensor` $\sqsubseteq$ `(measuresProperty only Property)`.

This modular extraction ensures that each reuse dimension is evaluated independently. To focus the analysis on meaningful reuse, the pipeline distinguishes between core ontology namespaces (e.g., `rdf`, `rdfs`, and `owl`) and domain-specific namespaces (e.g., SAREF [35], FIESTA [36]). Elements originating from core namespaces are filtered out. *Example:* a triple such as `rdfs:label` is excluded, whereas `saref:Sensor` is retained. This ensures that reuse metrics remain focused on domain-specific axioms.

4.5. Reuse Metrics Computation

For each feature set, reuse is computed as a normalised alignment between the reference ontology and the generated ontology:

$$\text{Reuse}(X) = \frac{1}{|X_r|} \sum_{x \in X_r} \max_{y \in X_g} \text{Sim}(x, y) \tag{3}$$

where X_r and X_g denote the sets extracted from the reference and generated ontologies, respectively, and $\text{Sim}(x, y) \in [0, 1]$ measures the similarity between elements x and y.

The similarity function is defined as:

$$\text{Sim}(x, y) = \begin{cases} 1, & \text{if } x = y \\ s(x, y), & \text{otherwise} \end{cases} \tag{4}$$

where $s(x, y)$ captures semantic similarity.

The reuse metrics are defined as follows:

- **Lexical Reuse (LR)**:

$$LR = \text{Reuse}(T) \tag{5}$$

This captures lexical alignment, allowing both exact and semantically similar matches. *Example:* overlap of class/property names such as `Sensor, Temperature`.

- **Structural Reuse (SR)**:

$$SR = \alpha\,\text{Reuse}(H) + (1-\alpha)\,\text{Reuse}(P) \tag{6}$$

This captures reuse of subclass hierarchies and property relations. *Example:* partial reuse of subclass hierarchies and property connections.

- **Logical Constraint Preservation (LC)**:

$$LC = \text{Reuse}(C) \times LV \tag{7}$$

Where $LV = 1$ if no unsatisfiable classes are detected via automated reasoning, and $LV = 0$ otherwise. Example: if constraints are reused but introduce inconsistency, then LC = 0.

- **Reuse depth: (RD)**:

$$RD = \gamma_1 LR + \gamma_2 SR + \gamma_3 LC \tag{8}$$

Aggregates these dimensions with $\sum_{i=1}^{4} \gamma_i = 1$. We assign higher weights to lexical and structural reuse ($\gamma_1 = \gamma_2 = 0.3$) than to logical constraints ($\gamma_3 = 0.2$), reflecting their greater importance in capturing meaningful ontology reuse. All metrics are normalized to the interval $[0, 1]$, enabling consistent comparison across ontologies.

4.6. Visualization and Analysis

The pipeline supports batch evaluation of multiple LLM-generated ontologies. Each ontology in a specified folder is evaluated against a reference ontology, and the computed metrics (LR, SR, LC, RD) are stored in a CSV file for subsequent analysis. Logical consistency is automatically verified using the Pellet reasoner via `Owlready2` integration. This automation enables reproducible, large-scale evaluation of ontology reuse across models and prompting strategies.

5. Experiment Results

To systematically analyse the results of the reuse of domain ontologies in LLM-generated ontologies, we developed a Python-based automated pipeline. Initially, ChatGPT attempted to create a base code. The pipeline operationalises our reuse framework. The automated pipeline provides a systematic and reproducible framework to quantify how well LLM-generated ontologies reuse domain-specific knowledge. By incorporating namespace filtering, modular feature extraction, and logical consistency checks, it produces metrics that are both meaningful and interpretable for comparing different ontology generation approaches.

To contextualise the obtained reuse scores with respect to the role of the reference ontologies in this study. The objective of the experiment is not to reproduce or reconstruct the reference ontologies, but rather to analyse the extent to which LLMs reuse existing ontologies when tasked with creating new ontologies in a given domain. This experiment assumes that the evaluated LLMs have prior knowledge of the reference ontologies, given their availability in publicly accessible sources. Future work will explicitly investigate this assumption by incorporating controlled assessments and experiments that distinguish between prior knowledge and reuse driven by the prompts.

The ontology generation task is open-ended and allows models to introduce new concepts and relations that may not be present in the reference ontologies. The reference ontologies serve as reuse targets rather than complete specifications of the domain. As a result, perfect reuse scores (i.e., values approaching 1.0) are not theoretically expected. The reported metrics should be interpreted as relative indicators of reuse behaviour across prompting strategies.

5.1. Results: SAREF Ontology

Table 1 presents the reuse metrics obtained when evaluating generated ontologies against the SAREF ontology. Figure 3 visualises the reuse depth scores across different prompting strategies. The results indicate that ontology reuse across all evaluated models is generally low. Lexical reuse represents the primary form of reuse observed. GPT-4.1 achieved the highest lexical reuse score (0.050) in the reuse-oriented prompt setting, indicating partial reuse of SAREF terminology. However, structural reuse was largely absent across most generated ontologies. Only the structured GPT-4.1 configuration demonstrated limited structural reuse (SR = 0.0189). This suggests that providing explicit structural guidance within prompts can encourage models to partially replicate ontology hierarchy patterns.

Table 1
Ontology reuse evaluation results using the SAREF ontology as reference

Prompt and LLM Type	LR	SR	LC	RD
Reuse-Oriented GPT-4o-mini	**0.0505**	0.0000	0.0000	0.0101
Reuse-Oriented Qwen2.5-7B	0.0101	0.0000	0.0000	0.0020
Reuse-Oriented Llama-3.1-8B	0.0100	0.0000	0.0000	0.0000
Reuse-Oriented GPT-4.1	0.0200	0.0189	0.0000	0.0170
Structured Llama-3.1-8B	0.0101	0.0000	0.0000	0.0020
Structured GPT-4o-mini	0.0101	0.0000	0.0000	0.0020
Structured Qwen2.5-7B	0.0101	0.0000	0.0000	0.0020
Structured GPT-4.1	0.0101	0.0000	0.0000	0.0020
Unstructured Llama-3.1-8B	0.0000	0.0000	0.0000	0.0000
Unstructured GPT-4o-mini	0.0000	0.0000	0.0000	0.0000
Unstructured Qwen2.5-7B	0.0000	0.0000	0.0000	0.0000
Unstructured GPT-4.1	0.0300	0.0000	0.0000	0.0010

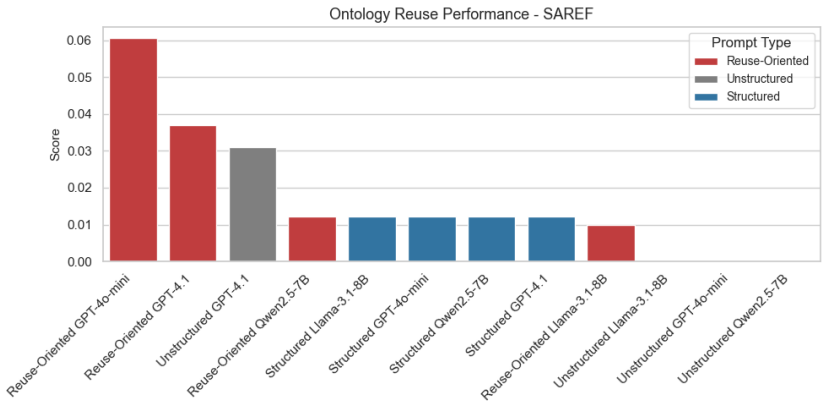

Figure 3: Reuse depth comparison for SAREF-based evaluation.

5.2. Results: FIESTA-IoT Ontology

Table 2 presents the results obtained when using the FIESTA-IoT ontology as the reference ontology. The results show that reuse scores are significantly lower when evaluated against the FIESTA ontology. Lexical reuse values remain below 0.01 across all configurations. This suggests that LLMs rarely reproduce terminology from the FIESTA-IoT ontology when generating ontologies without explicit guidance. A comprehensive explanation of these observations would require further investigation into the internal mechanisms of how LLMs generate output, which falls outside the scope of the present study. No structural reuse was detected in any generated ontology. Consequently, reuse depth values remain extremely low.

Table 2
Ontology reuse evaluation results against the FIESTA-IoT ontology

Prompt and LLM Type	LR	SR	LC	RD
Reuse-Oriented GPT-4.1	**0.0097**	0.0000	0.0000	0.0019
Reuse-Oriented Llama-3.1-8B	0.0058	0.0000	0.0000	0.0012
Reuse-Oriented Qwen2.5-7B	0.0058	0.0000	0.0000	0.0012
Reuse-Oriented GPT-4o-mini	0.0060	0.0000	0.0000	0.0020
Structured GPT-4.1	0.0039	0.0000	0.0000	0.0008
Structured Llama-3.1-8B	0.0058	0.0000	0.0000	0.0012
Structured Qwen2.5-7B	0.0058	0.0000	0.0000	0.0012
Structured GPT-4o-mini	0.0058	0.0000	0.0000	0.0012
Unstructured GPT-4.1	0.0000	0.0000	0.0000	0.0000
Unstructured Llama-3.1-8B	0.0000	0.0000	0.0000	0.0000
Unstructured Qwen2.5-7B	0.0039	0.0000	0.0000	0.0008
Unstructured GPT-4o-mini	0.0000	0.0000	0.0000	0.0020

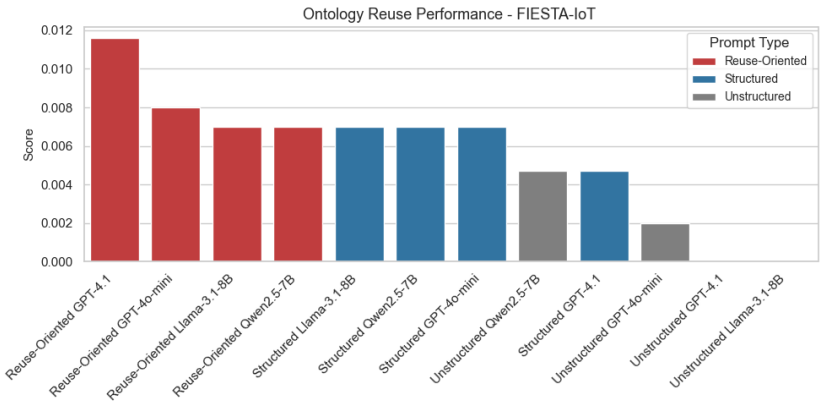

Figure 4: Reuse depth comparison for FIESTA-based evaluation.

Figure 4 illustrates the reuse depth distribution for the FIESTA evaluation.

Following the initial experiments, which suggested that reuse-oriented prompting can support deeper forms of ontology reuse, we conducted a more focused experiment using the same prompting strategy across multiple runs and models. In this phase, the reuse-oriented prompt was executed five times on the four LLMs, using the FIESTA-IoT ontology as the reference ontology. This ontology is particularly suitable for the analysis, as it was manually developed with the explicit goal of reusing and aligning existing ontologies. The evaluation considers structural characteristics, including the number of classes, object properties, data properties, and triples, as well as reuse metrics. Lexical reuse (LR) captures exact term overlap, while semantic lexical reuse (SLR) measures semantic similarity between lexical terms. Structural reuse (SR) captures the reuse of hierarchical and relational structures. The evaluation also reports precision, recall, and F1 scores to assess overall alignment with the reference ontology.

As shown in table 3 the results show a consistent pattern across models. LR and F1 scores are generally low, indicating limited direct reuse of reference terms, while SLR is higher, especially for GPT-based models and Qwen. GPT 4.1 shows some variability with occasional higher scores, whereas GPT 4o mini is more stable but lower. In contrast, Llama 3.1 and Qwen 2.5 often produce weakly structured outputs, leading to near-zero scores. SR remains low across all models, suggesting limited reuse of ontology relationships. Overall, reuse-oriented prompting encourages semantic similarity but does not consistently achieve true ontology reuse.

Table 3: Evaluation Results Across Models

Generated Ontology	#Classes	#Obj. Props	#Data Props	#Triples	LR	SLR	SR	Precision	Recall	F1-score
gpt-4.1_1.ttl	7	7	5	83	0.0000	0.3534	0.0000	0.0000	0.0000	0.0000
gpt-4.1_2.ttl	7	9	7	117	0.0000	0.3699	0.0007	0.0000	0.0000	0.0000
gpt-4.1_3.ttl	10	8	5	107	0.0104	0.4273	0.0031	0.5000	0.0104	0.0204
gpt-4.1_4.ttl	7	9	6	94	0.0000	0.3696	0.0007	0.0000	0.0000	0.0000
gpt-4.1_5.ttl	9	9	5	111	0.0000	0.3972	0.0000	0.0000	0.0000	0.0000
gpt-4o-mini_1.ttl	6	4	3	72	0.0063	0.4134	0.0024	0.5000	0.0063	0.0124
gpt-4o-mini_2.ttl	6	3	3	70	0.0063	0.4163	0.0017	0.5000	0.0063	0.0124
gpt-4o-mini_3.ttl	0	5	3	84	0.0000	0.0000	0.0021	0.0000	0.0000	0.0000
gpt-4o-mini_4.ttl	0	0	0	90	0.0000	0.0000	0.0017	0.0000	0.0000	0.0000
gpt-4o-mini_5.ttl	6	5	4	89	0.0042	0.4087	0.0017	0.3333	0.0042	0.0082
Llama-3.1-8B-Instruct1.ttl	0	0	0	35	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Llama-3.1-8B-Instruct2.ttl	0	0	0	78	0.0000	0.0000	0.0014	0.0000	0.0000	0.0000
Llama-3.1-8B-Instruct3.ttl	0	6	4	97	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Llama-3.1-8B-Instruct4.ttl	0	0	0	57	0.0000	0.0000	0.0007	0.0000	0.0000	0.0000
Llama-3.1-8B-Instruct5.ttl	4	4	4	40	0.0042	0.3697	0.0007	0.5000	0.0042	0.0083
Qwen2.5-7B-Instruct_reuse1.ttl	0	0	0	79	0.0000	0.0000	0.0024	0.0000	0.0000	0.0000
Qwen2.5-7B-Instruct_reuse2.ttl	7	5	7	101	0.0042	0.4163	0.0014	0.2857	0.0042	0.0082
Qwen2.5-7B-Instruct_reuse3.ttl	5	2	3	61	0.0042	0.3947	0.0014	0.4000	0.0042	0.0083
Qwen2.5-7B-Instruct_reuse4.ttl	11	3	3	97	0.0042	0.4214	0.0014	0.1818	0.0042	0.0082
Qwen2.5-7B-Instruct_reuse5.ttl	6	4	3	66	0.0042	0.3967	0.0014	0.3333	0.0042	0.0082

6. Conclusion

The experimental results reveal several important observations regarding ontology generation with LLMs. First, lexical reuse is the dominant form of reuse across all models and prompting strategies. This indicates that LLMs are capable of reproducing isolated domain terms but rarely replicate deeper ontological structures. Second, structural and semantic reuse are largely absent. Most generated ontologies failed to reproduce subclass hierarchies, domain-range relationships, or logical axioms from the reference ontologies. This finding suggests that current LLMs primarily generate ontologies through surface-level pattern synthesis rather than through reuse of formal ontology structures. Third, prompting strategies have a measurable impact on reuse behaviour. Structured prompts slightly improved structural reuse and semantic reuse for GPT-4.1, demonstrating that explicit instructions about ontology structure can encourage deeper modelling patterns. Finally, differences between the two reference ontologies highlight the importance of ontology vocabulary visibility. SAREF achieved slightly higher reuse scores due to its wider recognition and clearer domain terminology. In contrast, FIESTA reuse remained minimal across the experiments.

As this study represents a first attempt to define metrics for quantifying ontology reuse in LLM-generated outputs, the findings must be interpreted within certain methodological constraints. The proposed measures provide useful insights into lexical, structural, and semantic reuse patterns, they also have limitations. In particular, the metrics rely on axiom-level comparisons and may not fully capture more subtle forms of semantic equivalence. Therefore, the results should be interpreted as indicative rather than exhaustive, and future work is needed to validate them across a broader range of ontologies, domains, and generation settings.

Overall, the results indicate that while LLMs can assist with ontology engineering, their ability to reuse existing ontologies remains limited without explicit guidance or integration mechanisms. These findings highlight the need for improved prompting strategies and reuse-aware generation methods. Investigating these approaches represents an important direction for future research aimed at developing reuse-aware generation methods and improving the integration of LLMs with semantic knowledge bases.

Declaration on Generative AI

During the preparation of this work, the authors used the ChatGPT model (based on OpenAI's GPT-5.3 architecture) in order to generate an initial Python automation code and a preliminary version of the code used for the evaluation of the ontologies generated by the LLM. The generated code served as a starting point and was subsequently reviewed, modified, and adapted by the author(s) as needed.

Acknowledgment

This work has partially received funding from the European Union's Horizon Europe research and innovation programme under grant agreement RESONANCE No. 101096200. The opinions expressed are those of the authors and do not reflect those of the sponsors.

References

[1] H. S. Pinto, J. P. Martins, Reusing ontologies, in: AAAI Technical Report SS-00-03, Spring Symposium on Ontology Management, AAAI Press, Palo Alto, CA, 2000, pp. 1–? URL: https://cdn.aaai.org/Symposia/Spring/2000/SS0003/SS00-03012.pdf.

[2] D. Lonsdale, D. W. Embley, Y. Ding, L. Xu, M. Hepp, Reusing ontologies and language components for ontology generation, Data & Knowledge Engineering 69 (2010-04) 318–330. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169023X09001153. doi:10.1016/j.datak.2009.08.003.

[3] OBO Foundry, The obo foundry: Coordinated evolution of ontologies to support biomedical data integration, https://obofoundry.org, 2026. Accessed: 2026-03-12.
[4] National Center for Biomedical Ontology, Bioportal: Ontology repository and browser, https://bioportal.bioontology.org, 2026. Accessed: 2026-03-12.
[5] Linked Open Vocabularies (LOV), A registry of rdf vocabularies for the semantic web, https://lov.linkeddata.es, 2026. Accessed: 2026-03-12.
[6] Stanford Center for Biomedical Informatics Research, Protégé: A free, open-source ontology editor and framework for building intelligent systems, https://protege.stanford.edu, 2026. Accessed: 2026-03-12.
[7] E. Simperl, Reusing ontologies on the Semantic Web: A feasibility study, Data & Knowledge Engineering 68 (2009-10) 905–925. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169023X0900007X. doi:10.1016/j.datak.2009.02.002.
[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.
[9] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, arXiv preprint arXiv:2204.02311 (2022).
[10] OpenAI, Gpt-4 technical report, OpenAI, 2023. https://openai.com/research/gpt-4.
[11] D. Garijo, M. Poveda-Villalón, E. Amador-Domínguez, Z. Wang, R. García-Castro, Óscar Corcho, Llms for ontology engineering: A landscape of tasks and benchmarking challenges, in: Proceedings of the Special Session on Harmonising Generative AI and Semantic Web Technologies (HGAIS) at ISWC 2024, volume 3953, CEUR Workshop Proceedings, 2024, pp. 1–? URL: https://ceur-ws.org/Vol-3953/364.pdf, cC BY 4.0.
[12] J. Plu, O. Moreno Escobar, E. Trouillez, A. Gapin, R. Troncy, A comprehensive benchmark for evaluating llm-generated ontologies, in: Proceedings of the Special Session on Harmonising Generative AI and Semantic Web Technologies (HGAIS 2024) co-located with the 23rd International Semantic Web Conference (ISWC 2024), volume 3953 of *CEUR Workshop Proceedings*, CEUR-WS.org, Baltimore, MD, USA, 2025, pp. 1–? URL: https://ceur-ws.org/Vol-3953/362.pdf.
[13] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220. doi:10.1006/knac.1993.1008, defines ontology as an explicit specification of conceptualization.
[14] R. Studer, V. R. Benjamins, D. Fensel, Knowledge engineering: principles and methods, Data & Knowledge Engineering 25 (1998) 161–197.
[15] M. D. Wilkinson, M. Dumontier, g.-i. family=Aalbersberg, given=IJsbrand Jan, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. Da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 'T Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. Van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. Van Der Lei, E. Van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons, The FAIR Guiding Principles for scientific data management and stewardship, Scientific Data 3 (2016-03-15) 160018. URL: https://www.nature.com/articles/sdata201618. doi:10.1038/sdata.2016.18.
[16] M. Uschold, P. Clark, M. Healy, K. Williamson, S. Woods, An experiment in ontology reuse, 1998. URL: https://ksi.cpsc.ucalgary.ca/KAW/KAW98/uschold/, accessed: July 22, 2025.
[17] M. Katsumi, M. Grüninger, What is ontology reuse?, in: Frontiers in Artificial Intelligence and Applications, IOS Press, 2016, pp. 1–? URL: https://www.medra.org/servlet/aliasResolver?alias=iospressISBN&isbn=978-1-61499-659-0&spage=9&doi=10.3233/978-1-61499-660-6-9. doi:10.3233/978-1-61499-660-6-9.
[18] N. F. Noy, D. L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory, 2001.

[19] EU RESONANCE Project, Resonance ontology (sri4all, sri4ev modules), https://github.com/EU-Resonance-Software/SRI, 2024. Accessed: 2026-04-16.
[20] V. Authors, Saref4ener: An extension of saref for the energy domain, Semantic Web / IoT literature (2020). SAREF4ENER extends SAREF for smart energy systems.
[21] H. Babaei Giglou, J. D'Souza, S. Auer, LLMs4OL: Large Language Models for Ontology Learning, in: T. R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, J. Li (Eds.), The Semantic Web – ISWC 2023, volume 14265, Springer Nature Switzerland, 2023, pp. 408–427. URL: https://link.springer.com/10.1007/978-3-031-47240-4_22. doi:10.1007/978-3-031-47240-4_22.
[22] N. Fathallah, S. Staab, A. Algergawy, LLMs4Life: Large Language Models for Ontology Learning in Life Sciences, 2024. URL: https://arxiv.org/abs/2412.02035. doi:10.48550/ARXIV.2412.02035.
[23] N. Fathallah, A. Das, S. D. Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, NeOn-GPT: A Large Language Model-Powered Pipeline for Ontology Learning, in: The Semantic Web: ESWC 2024 Satellite Events, volume 15344, Springer Nature Switzerland, 2025, pp. 36–50. URL: https://link.springer.com/10.1007/978-3-031-78952-6_4. doi:10.1007/978-3-031-78952-6_4.
[24] H. T. Mai, C. X. Chu, H. Paulheim, Do LLMs Really Adapt to Domains? An Ontology Learning Perspective, in: G. Demartini, K. Hose, M. Acosta, M. Palmonari, G. Cheng, H. Skaf-Molli, N. Ferranti, D. Hernández, A. Hogan (Eds.), The Semantic Web – ISWC 2024, volume 15231, Springer Nature Switzerland, 2025, pp. 126–143. URL: https://link.springer.com/10.1007/978-3-031-77844-5_7. doi:10.1007/978-3-031-77844-5_7.
[25] Z. Qiang, W. Wang, K. Taylor, Agent-OM: Leveraging LLM Agents for Ontology Matching, 2023. URL: https://arxiv.org/abs/2312.00326. doi:10.48550/ARXIV.2312.00326.
[26] M. Val-Calvo, M. Egaña Aranguren, J. Mulero-Hernández, G. Almagro-Hernández, P. Deshmukh, J. A. Bernabé-Díaz, P. Espinoza-Arias, J. L. Sánchez-Fernández, J. Mueller, J. T. Fernández-Breis, OntoGenix: Leveraging Large Language Models for enhanced ontology engineering from datasets, Information Processing & Management 62 (2025-05) 104042. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306457324004011. doi:10.1016/j.ipm.2024.104042.
[27] P. Mateiu, A. Groza, Ontology engineering with Large Language Models, 2023-07-31. URL: http://arxiv.org/abs/2307.16699. doi:10.48550/arXiv.2307.16699. arXiv:2307.16699.
[28] A. S. Lippolis, M. J. Saeedizade, R. Keskisärkkä, S. Zuppiroli, M. Ceriani, A. Gangemi, E. Blomqvist, A. G. Nuzzolese, Ontology Generation using Large Language Models, 2025-03-07. URL: http://arxiv.org/abs/2503.05388. doi:10.48550/arXiv.2503.05388. arXiv:2503.05388.
[29] R. Alharbi, V. Tamma, F. Grasso, T. Payne, An experiment in retrofitting competency questions for existing ontologies, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, Sac '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 1650–1658. doi:10.1145/3605098.3636053.
[30] R. D. Team, Rdflib: A python library for working with rdf, 2024. URL: https://github.com/RDFLib/rdflib, accessed: 2026-04-16.
[31] OpenAI, Gpt-4o mini, https://platform.openai.com/docs/models/gpt-4o-mini, 2024. Accessed: 2026-04-14.
[32] OpenAI, Gpt-4.1, https://platform.openai.com/docs/models/gpt-4-1, 2025. Accessed: 2026-04-14.
[33] Meta AI, Llama 3.1 8b instruct, https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct, 2024. Accessed: 2026-04-14.
[34] Alibaba Cloud, Qwen2.5-7b-instruct, https://huggingface.co/Qwen/Qwen2.5-7B-Instruct, 2024. Accessed: 2026-04-14.
[35] ETSI, Saref ontology, https://saref.etsi.org/, 2019. Accessed: 2026-04-14.
[36] FIESTA-IoT Consortium, Fiesta-iot ontology, https://fiesta-iot.eu/, 2018. Accessed: 2026-04-14.

A. Appendix

This appendix provides the full prompts used in the experiments. The same prompts were applied across all evaluated LLMs.

A.1. Unstructured Prompt

```
Generate an ontology for the domain of <DOMAIN>.
The ontology should capture the main concepts and relationships relevant to this domain.
```

A.2. Structured Prompt

```
Generate an ontology for the domain of <DOMAIN>.
Requirements:
- Identify and define the core classes (key concepts in the domain)
- Define object properties to represent relationships between classes
- Define data properties to represent attributes of entities
- Include a clear and coherent class hierarchy
- Use meaningful and consistent naming conventions
Output format:
- Provide the ontology in Turtle (RDF) syntax
- Ensure the syntax is valid and well-structured
```

A.3. Reuse-Oriented Prompt

```
Generate an ontology for the domain of <DOMAIN>.
Reuse Strategy:
- Reuse existing ontologies, vocabularies, or standards whenever appropriate
(e.g., <DOMAIN ONTOLOGY>)
- Prefer reusing existing classes and properties over creating new ones
- Clearly align or map reused elements where applicable
Modeling Requirements:
- Define classes, object properties, and data properties
- Ensure a clear class hierarchy and well-defined relationships
- Avoid redundancy and unnecessary duplication of concepts
Output format:
- Provide the ontology in Turtle (RDF) syntax
- Ensure correctness and consistency of reused terms
```