Ontology Learning with LLMs: A Benchmark Study on Axiom Identification

Journal Title
XX(X):2–19

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

Roos M. Bakker[1,2], Daan L. Di Scala[1,3], Maaike H.T. de Boer[1], Stephan A. Raaijmakers[1,2]

Abstract

Ontologies are an important tool for structuring domain knowledge, but their development is a complex task that requires significant modelling and domain expertise. Ontology learning, aimed at automating this process, has seen advancements in the past decade with the improvement of Natural Language Processing techniques, and especially with the recent growth of Large Language Models (LLMs). This paper investigates the challenge of identifying axioms: fundamental ontology components that define logical relations between classes and properties. In this work, we introduce an Ontology Axiom Benchmark *OntoAxiom*, and systematically test LLMs on that benchmark for axiom identification, evaluating different prompting strategies, ontologies, and axiom types. The benchmark consists of nine medium-sized ontologies with together 17.118 triples, and 2.771 axioms. We focus on subclass, disjoint, subproperty, domain, and range axioms. To evaluate LLM performance, we compare twelve LLMs with three shot settings and two prompting strategies: a Direct approach where we query all axioms at once, versus an Axiom-by-Axiom (AbA) approach, where each prompt queries for one axiom only. Our findings show that the AbA prompting leads to higher F1 scores than the direct approach. However, performance varies across axioms, suggesting that certain axioms are more challenging to identify. The domain also influences performance: the FOAF ontology achieves a score of 0.642 for the subclass axiom, while the music ontology reaches only 0.218. Larger LLMs outperform smaller ones, but smaller models may still be viable for resource-constrained settings. Although performance overall is not high enough to fully automate axiom identification, LLMs can provide valuable candidate axioms to support ontology engineers with the development and refinement of ontologies.

Keywords

Ontology Learning, Axiom Identification, Large Language Models, Ontology Engineering, Ontology Evolution

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

1 Introduction

Knowledge graphs and formal ontologies are increasingly vital in modern information systems and AI-based decision support systems. These formal representations provide a transparent and explainable complement to black-box machine learning models. Ontologies define type entities and relations that describe and structure aspects of a domain of interest [1], enabling reasoning and interoperability over various data sources. However, constructing and maintaining such formal models is a complex and resource-intensive task, requiring manual effort not only for the initial development but also for updates to ensure alignment with evolving applications. Data-driven techniques can help automate this process by identifying ontology elements from text or other domain sources. In this paper, we investigate the ability of language models to automatically identify axioms in ontologies.

Early work on automatically identifying ontologies, also called ontology learning, made the distinction between different components of the ontology, such as terms, taxonomical relations, and axioms [2,3]. An example of an axiom is the disjoint relation, where two concepts are disjoint if no individual can simultaneously belong to both classes (e.g., a mountain cannot also be a river, and vice versa). An early overview of Buitelaar et al. [4] on ontology learning from text identified differences in complexity between these tasks and visualises it in the ontology learning layer cake, as shown in Figure 1. At the bottom of the cake is the automatic learning of terms, considered the least complex task. It builds up to synonyms, concepts (consisting of intensions I, extensions E and lexicon L) and taxonomical concept hierarchies. At the top layer are rules and axioms, with examples such as disjoint, domain, and range relations. A domain axiom specifies the allowed subject of a relation. For instance, the relation FLOWS_THROUGH has a domain of RIVER. A range axiom specifies the allowed object of a relation, such as CAPITAL_OF, which has a range of COUNTRY. Early research mainly focused on the lower levels by creating rule-based solutions using syntactical patterns, with the results needing manual adjustments to produce ontologies. In the last decade, the field of ontology learning and engineering has advanced rapidly, driven by machine learning and Natural Language Processing (NLP) techniques [5]. Recently, Large Language Models (LLMs) have shown promise in supporting ontology learning tasks [6,7], especially when combined with human expertise for parts of the ontology development process [8].

Much of the progress in ontology learning so far has concentrated on the lower levels of the ontology learning layer cake, such as term and taxonomy extraction, while relatively little attention has been paid to the automated identification of axioms. Unlike terms and relations, axioms are often implicit and abstract, making them difficult to identify automatically [9]. Additionally, the flexibility in how axioms can be modelled further complicates the task. Modelling axioms is also a challenging task for humans, as previous studies have pointed out [10,11]. Despite these challenges, axioms are important to include

[1] Netherlands Organisation for Applied Scientific Research (TNO)
[2] Leiden University Centre for Linguistics (LUCL)
[3] Utrecht University

Corresponding author:
Roos M. Bakker, TNO Anna van Buerenplein 1, Den Haag, 2595 DA, NL.
Email: roos.bakker@tno.nl

Prepared using sagej.cls

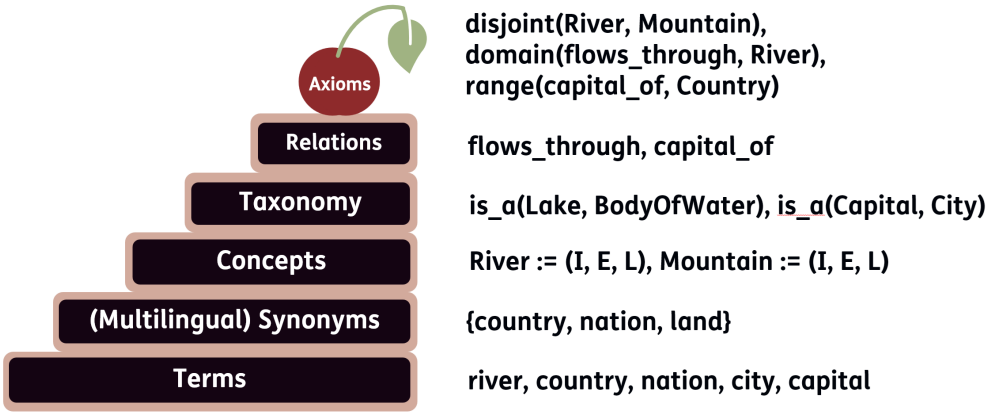

Figure 1. Ontology learning layer cake (Figure adapted from[4]).

since they add to the expressivity and logical consistency of an ontology, enabling inference, validation, and richer reasoning capabilities[12,13]. Being able to automatically identify them, or semi-automatically by giving suggestions, would provide valuable assistance to an ontology developer. Given the broad capabilities of LLMs in handling abstract reasoning and contextual information and their promising results in related tasks, we investigate their ability to identify axioms in ontologies. For ontology engineers, automating such parts of ontology development could lighten the workload and accelerate the process.

In this paper, we investigate automatic axiom identification by LLMs. Specifically, we investigate how well LLMs can infer axioms in ontologies. To this end, we introduce the Ontology Axiom Benchmark *OntoAxiom* that contains nine ontologies and a total of 17.118 triples and 2.771 axioms. For the experiment with LLMs, we compare two prompting strategies: a Direct approach, where all axioms are predicted at once, and an Axiom-by-Axiom (AbA) approach, where separate prompts are used per axiom type. Our experiments include twelve LLMs and three shot settings. We evaluate the results using precision, recall, and F1 scores. We provide full results and additional data in an open source repository*.

This paper is structured as follows: We start by discussing related work on ontology learning, recent developments in applying LLMs to this field, and the role of axioms. In Section 3, we introduce the Ontology Axiom Benchmark by discussing axiom and ontology selection criteria. In Section 4, we outline our experimental methodology, covering the LLMs, prompt strategies, and evaluation methods. Section 5 presents the results of our experiments. Finally, we provide a discussion of our findings and their implications in Section 6, and we conclude our work in Section 7 and discuss future work.

*https://gitlab.com/ontologylearning/axiomidentification

2 Related Work

In this section, we will describe related work on ontology learning, LLMs, and axiom identification. In Section 2.1, we will briefly sketch a history of the field of automatically, or semi-automatically building ontologies and similar models such as knowledge graphs. In Section 2.2, we will discuss recent advances in using LLMs for ontology learning tasks. Finally, in Section 2.3, we zoom in on related work on automatic axiom identification.

2.1 Ontology Learning

Creating ontologies is a time-consuming task which requires the effort of developers and domain experts to properly capture concepts within a domain. The field of ontology learning aims to automate (parts of) this process. Early approaches mostly applied rule-based frameworks[4], with more recent approaches adopting machine learning techniques[9]. Giunchiglia et al.[14] make a different distinction in complexity. They introduce lightweight ontologies that contain no axioms and complex relations. Heavyweight ontologies, on the other hand, contain more axioms and complex relations[15]. The lightweight ontologies were easier to automatically create from text[16].

With the rise of machine learning techniques and increased computing power, statistical approaches have become more effective in ontology learning. Asim et al.[5] discuss techniques such as co-occurrences, clustering, and word embeddings, comparing linguistic and statistical methods. They also differentiate between term extraction, generally considered the easier task, and relation extraction, considered more complex. Recently, Amalki et al.[17] have explored deep-learning techniques such as Recurrent Neural Networks (RNNs), Graph Neural Networks (GNNs) and (knowledge) graph embeddings for ontology learning, including the application of Large Language Models across various domains. They mention an increasing trend in the application of LLMs in the field, as well as a shift towards the use of generalizable, large-scale models. They found that, despite the clear evolution of the field in recent years, challenges remain for smaller domains with less data, particularly in the creation of comparative studies.

2.2 LLMs for ontology learning

The key technology most current LLMs use is the transformer architecture, which consists of two interacting models, an encoder and a decoder[18]. The LLM can be trained end-to-end (such as in Flan-T5[19]), or using encoder-only architectures (for example BERT[20]) or decoder-only (for example GPT[21,22]) models.

The potential of LLMs has led to significant progress in ontology learning by enabling more accurate and scalable extraction of concepts and relations from unstructured text[17]. Examples of popular LLMs in ontology learning are GPT models, Claude, Mistral, BERT, FLAN-T5, Bloom, and RoBERTa[17]. Babaei et al.[6] compare the performance of several (Large) Language Models, namely BERT (encoder-only), BLOOM, Llama, GPT-3, GPT-3.5, GPT-4 (all decoder-only), BART and Flan-T5 on the task of Term Typing, Taxonomy Discovery and Non-Taxonomic Relation Extraction. Results show that, for term typing, domain-independent models with large-scale parameters such as GPT and Bloom outperform the other models. For taxonomy discovery, GPT models also outperform the other models, and there is a 10% performance gap with the open source models. For non-taxonomic relations, Flan was the best model (in 2023). Another result is that the models with the biggest number of parameters have the best

results over all tasks, and that, at least for Flan, instruction tuning increases performance on the different tasks.

Beyond step-by-step approaches on different subtasks, researchers have also explored end-to-end ontology learning, where multiple subtasks, such as term and relation extraction, are learned jointly in a single step[7,23,24]. These efforts have primarily focused on extracting concepts, relations, and taxonomic structures[7]. A key distinction among these works lies in both the prompting techniques used and the type of input from which ontologies are generated. Inputs range from full-domain texts[23] to more structured sources like competency questions and short user stories[24]. Prompting strategies vary as well, including zero-shot and few-shot approaches (where a small number of labelled examples is included in the prompt), as well as methods that decompose the task, for example, by prompting separately for each competency question[25]. Ciatto et al.[26] take a different approach by populating an existing ontology by instantiating classes and properties with LLMs, thereby showing the feasibility of using an LLM as an oracle for ontology population.

Many of these studies are limited to restricted domains, such as music or theatre, where LLMs can use their patterns learned during training to generate the content of the ontologies[24]. While these controlled settings provide useful insights, they do not always translate to more specialised or real-world applications[17]. For more complex domains, additional domain-specific information can be incorporated, such as news articles[23] or manually curated domain descriptions[27].

Despite these advancements, significant challenges remain. The lack of standardised benchmarks, the absence of good statistical paradigms to evaluate these benchmarks, the abundance of errors in benchmarks and difficulties in evaluation make it hard to compare different approaches and track progress[7,17,24,28].

2.3 Axiom Identification

Axioms, being at the top of the ontology learning layer cake in Figure 1, are considered a challenging part of ontology learning because they require inferring abstract logical relations rather than surface-level associations.[4,29]. There are a few works that specifically focus on automatic axiom identification. Haase et al.[30] take a confidence based approach of ontology learning. For equivalence axioms, they employ a corpus based similarity metric to calculate a confidence score of equality. For disjoint, they implement a heuristic based on lexico-syntactic patterns. This is based on the evidence found for the disjointness in relation to the overall evidence of disjointness throughout all concepts.

Bühmann and Lehmann[31] mine frequent axiom patterns from over 1300 ontologies and validate their method by manually evaluating them. Their evaluation shows experts deeming 48.2% of the proposed axioms to be useful extensions to the ontology, indicating the potential usefulness of automated axiom identification for ontology engineering, with room for improvement.

Ballout et al.[32] propose a method for axiom learning based on similarity scores derived from ontological distances between concepts. They test these axiom similarity matrices for subClassOf and disjointWith axioms with various regression methods, including random forests and support vector regression.

More recently, LLMs are used to identify axioms from natural language. Mateiu and Groza[33] finetune a GPT-3 model to convert natural language into structured OWL Functional Syntax. Among other parts of the target ontology such as class subsumption, they include the taxonomical subclass (subClassOf), class disjointness (disjointWith) and class equivalence axioms in their approach. They furthermore prompt

GPT-3 to provide domain and range of extracted relations, as well as provide examples of subproperty (subPropertyOf), symmetric and asymmetric relations. However, their evaluation is limited to a brief qualitative assessment of selected examples, making it difficult to fully assess the potential of LLMs for axiom identification.

2.4 Summary

Based on the literature, LLMs have led to significant progress in ontology learning[7,17,23,24]. Research has focused on the extraction of concepts, relations and taxonomic structures, but very limited research has been done on axioms[17]. To the best of our knowledge, the only research done with axioms and LLMs is by Mateiu and Groza[33]. This paper explores the potential for the usage of LLMs for axiom extraction.

3 Ontology Axiom Benchmark

In this paper, we introduce the Ontology Axiom Benchmark *OntoAxiom*. We selected five axioms, which will be discussed in Section 3.1. Based on these axioms and additional ontology selection criteria, we select nine existing ontologies, which are discussed in Section 3.2.

3.1 Axiom Selection

Table 1. Summary of common ontology axioms, their corresponding description logic and a description of their meaning in natural language.

Axiom	D.Logic	Description
subClassOf	$C_1 \sqsubseteq C_2$	All the instances of one class C_1 are instances of another class C_2
disjointWith	$C_1 \sqsubseteq \neg C_2$	No instance of one class C_1 can be an instance of another class C_2
subPropertyOf	$P_1 \sqsubseteq P_2$	All resources related by one property P_1 are related by another property P_2
range	$\exists P.\top \sqsubseteq C$	The values of a property P are instances of class C
domain	$\top \sqsubseteq \forall P.C$	Any resource that has a given property P is an instance of class C

Formal axioms are an important part of ontologies, since they represent universally true knowledge and enforce logical constraints. This allows for reasoning and ensures consistency within an ontology[34]. To evaluate a representative range of axioms, we choose to include both OWL and RDFS axioms, and the choice is made to include both class axioms as well as property axioms. This leads us to focus on the following five well-known and often used axioms: RDFS:SUBCLASSOF, OWL:DISJOINTWITH, RDFS:SUBPROPERTYOF, RDFS:RANGE, and RDFS:DOMAIN. Table 1 shows these axioms, their corresponding description logic, and textual definitions. Together, these five axioms form an impactful toolkit to formally model domains with sufficient rules. We introduce the selected five axioms more in-depth in the rest of this section.

SubClass and SubProperty. The subclass (RDFS:SUBCLASSOF) and subproperty (RDFS: SUBPROPERTYOF) axioms establish hierarchical relationships between classes and properties, respectively. These axioms allow for inheritance of properties, abstraction and generalisation. An example for the subclass axiom is :TURTLE RDFS:SUBCLASSOF :ANIMAL, which allows to infer :TURTLE being an :ANIMAL. Similarly, for subproperty, an example would be :OWNSTURTLE RDFS:SUBPROPERTYOF :OWNSPET, which asserts that for any class with :OWNSTURTLE also implies :OWNSPET.

Disjoint. Complementing this, the disjoint OWL:DISJOINTWITH axiom introduces the constraint that asserts two classes cannot share instances, called disjointness. For example, :TURTLE OWL:DISJOINTWITH :DOG means that a :Turtle is not a :Dog, something which enforces clarity in distinction and prevents semantic ambiguity. Note that disjointness can also be defined through the OWL:ALLDISJOINTCLASSES axiom, which identifies classes that are all disjoint from one another. So, RDF:TYPE OWL:ALLDISJOINTCLASSES; OWL:MEMBERS (:DOG :CAT :TURTLE). shows that an instance can only be of DOGS, :CATS or :TURTLES, and not the others.

Domain and Range. Finally, the domain (RDFS:DOMAIN) and range (RDFS:RANGE) axioms further enrich the ontologies by constraining the subjects and objects that properties can relate to. Continuing the example, :OWNSTURTLE RDFS:DOMAIN :PERSON and :OWNSTURTLE RDFS:RANGE :TURTLE enable explicit typing, with any triple using :OWNSTURTLE implies that the subject must be a :PERSON and the object must be a TURTLE.

3.2 Ontology Selection

Table 2. Overview of ontologies in the Ontology Axiom Benchmark, with general information, information on class axioms and information on property axioms. The largest amounts are indicated with bold text.

	ERA	FOAF	GoodRel	gUFO	MO	NS	Pizza	SAREF	Time
triples	7021	631	1834	766	2141	362	1848	1151	1364
classes	50	15	37	50	60	35	**93**	31	20
properties	**459**	60	102	47	165	35	8	71	58
subClassOf	20	10	19	56	59	20	**80**	11	16
disjointWith	4	8	**256**	29	2	4	5	0	1
allDisjoint	0	0	0	6	0	0	**390**	0	0
subproperty	25	13	22	14	**81**	0	4	8	13
domain	**448**	55	96	47	132	1	3	52	55
range	**456**	55	96	43	128	1	4	48	55

To evaluate the capabilities of axiom identification, the benchmark needs to consist of representative existing ontologies. For this, we have collected ontologies that fit the following three criteria:

1. **Ontology Size**. The ontologies should be of adequate size to provide proper insight. Ontologies that are too small, while adding to different topics, might yield too little information during evaluation. On the other hand, ontologies can get very large, which is undesirable from a technical limitations standpoint. For this, we have decided to scope the ontology size between 15 and 100 classes, as well as a maximum of 500 properties.
2. **Axiom Amount.** The ontologies should have sufficient axioms to test on. In our scope, we are focusing on the SUBCLASS, DISJOINT, SUBPROPERTY, DOMAIN and RANGE axioms. Ideally, the ontology includes all of these axioms. However, in practice, this is often not the case. Therefore, we have set this criterion to at least 20 and at most 500 instances of one of the five axioms.
3. **Topic Coverage.** The benchmark should consist of a wide array of topics. Therefore, each selected ontology should introduce semantically different topics and relations when compared to other

ontologies in the dataset. Preferably, different types of ontologies are included, such as everyday items, basic descriptions and abstract concepts.

Based on the above criteria, we selected nine different mid-sized ontologies, which comprise the Ontology Axiom Benchmark. The nine ontologies are as follows:

- **Railways.** The ontology by the European Union Agency for Railways (ERA)[35] describes the European railway infrastructure, with concepts such as ERA:TRACK and ERA:MAGNETICBRAKEPREVENTION.
- **Persons.** The Friend-of-a-Friend ontology (FOAF)[36] is a classic example ontology which describes people, their activities, and their relationships. It describes concepts such as FOAF:GROUP and FOAF:ONLINECHATACCOUNT.
- **Products.** The GoodRelations ontology[37] is used to describe commodity products and services for web sales. It includes concepts such as GR:WARRANTYSCOPE and GR:PAYMENTMETHODCREDITCARD.
- **Foundational Knowledge.** gUFO[38] is a lightweight version of the Unified Foundational Ontology (UFO). It holds foundational concepts such as GUFO:NONSORTAL and GUFO:ENDURANT.
- **Music.** The Music Ontology[39] contains knowledge on the music domain, with concepts such as MO:SOLOMUSICARTIST and MO:RECORDING.
- **Safety.** The Nord Stream Pipeline ontology[40] holds knowledge extracted from a news item about the attack on the Nord Stream Pipeline. It contains concepts such as NS:GASSUPPLY and NS:STATEMENT.
- **Pizza.** The Pizza ontology is a classic example ontology about pizza, which holds concepts like :PIZZABASE and :SAUCETOPPING[41]. For this paper, we cleaned the Pizza ontology by removing the following contradicting and irrelevant educational example classes: :CHEESEYVEGETABLETOPPING, :SPICYPIZZAEQUIVALENT, :VEGETARIANPIZZAEQUIVALENT1, :VEGETARIANPIZZAEQUIVALENT2, :UNCLOSEDPIZZA, and :ICECREAM).
- **Smart Appliances.** SAREF[42] is an ontology for the smart applications domain, with concepts such as SAREF:DEVICE and SAREF:FEATUREOFINTEREST.
- **Time.** The OWL-Time ontology[43] describes temporal concepts, such as TIME: FRIDAY and TIME:DATETIMEDESCRIPTION.

An overview of the ontologies in the benchmark is provided in Table 2, where the number of triples, classes, properties, and different axioms is shown for each ontology. With this, the benchmark spans a variety of topics for a wide coverage: Railways, Products, Persons, Foundational Knowledge, Music, Safety, Pizza, Smart Appliances and Time. In total, the benchmark consists of 17.118 triples, totalling 2.771 axioms. The benchmark is made available through the open-source repository*.

4 Method

This paper explores the ability of LLMs to identify five types of axioms (shown in Table 1). We compare nine ontologies, which were detailed in Section 3. We test two different prompting approaches with three different shot settings, which are described in Section 4.1. We test twelve different LLMs, which are introduced in Section 4.2. Finally, we discuss the evaluation strategies in Section 4.3.

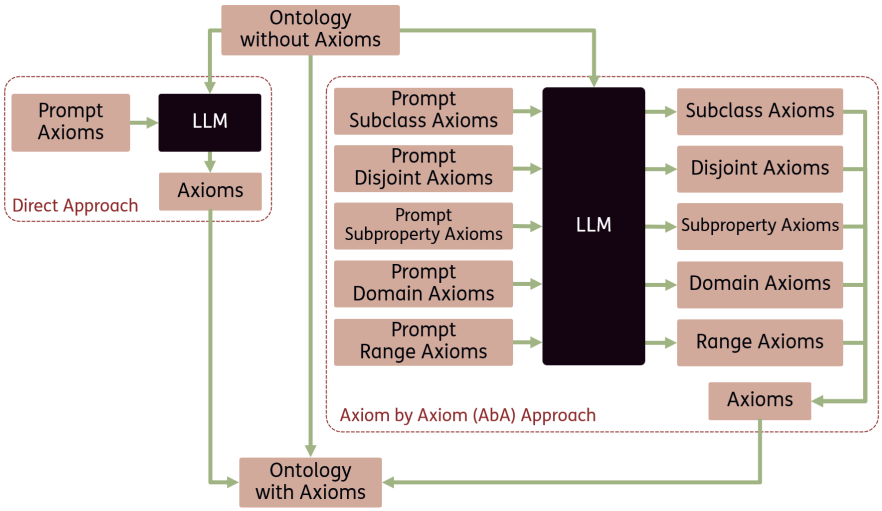

Figure 2. An overview of the two different approaches that we test for axiom identification. Left shows the Direct approach, right the Axiom by Axiom (AbA).

4.1 Axiom Identification Approaches

In our experiments, we test two approaches, as shown in Figure 2. The Figure shows at the top the start of the process with the input: the ontology without axioms. In the middle, the two different approaches are shown, which lead to the end of the process with the output: the ontology with axioms.

In the **Direct** approach, an LLM is prompted to identify all axioms at once. In the **Axiom-by-Axiom (AbA)** approach, the axiom identification task is decomposed into smaller steps. The LLM is prompted with separate prompts for each axiom, after which all axioms are aggregated.

For each of these approaches, we test three shot settings: zero-, one-, and five-shot prompts, an amount frequently chosen for a few-shot setting in previous work[44]. For the zero-shot prompt setting, no examples are included. For the one-shot setting, one labelled example for each axiom is included for the Direct approach, and one example for each separate AbA prompt is included. Finally, for the five-shot approach, five labelled examples are included. The four prompt templates are shown in Table 3: two shot settings for the Direct approach, and two for the AbA approach. For AbA, the prompts for the other axioms are similar to the one shown in the Table, but with different examples. All prompts and included examples can be found in our repository*.

4.2 LLMs

For our experiments, we selected language models based on whether 1) they are open source or proprietary, 2) what family they are from, 3) what size they are, and 4) their specialisation. Our goal was to ensure sufficient variety along these dimensions. All models in our selection are instruction-tuned, and for all models we use a temperature setting of 0.2 to ensure a more consistent output. An overview is provided in Table 4, which also describes the number of parameters. Since the proprietary models from OpenAI are not open source, their parameter counts are not publicly disclosed. We estimated the

Table 3. Prompt templates for both approaches and two shot settings.

Approach & Shot setting	Prompt Template
Direct Zero-shot	Identify axioms between these classes and properties. The class axioms are subclass and disjointness. The property axioms are subproperty, domain and range. Return the output as a JSON object with five keys: SUBCLASS, DISJOINT, SUBPROPERTY, DOMAIN, RANGE. Each key should map to a list of lists. {CLASSES}, {PROPERTIES}
Direct One-shot	Identify axioms between these classes and properties. These are examples: {subclass: [[Cat, Animal]], disjoint: [[Cat, Dog]], subproperty: [[hasMother, hasParent]], domain: [[Event, happenedAt]], range: [[happenedAt, Place]] } {CLASSES}, {PROPERTIES}
AbA Zero-shot (Subclass)	Identify SUBCLASS relations between these classes: {CLASSES}. Return an array of arrays, where each inner array contains the subclass first and the superclass second. {CLASSES}, {PROPERTIES}
AbA One-shot (Subclass)	Identify subclass relations between these classes. This is an example: {[[Cat, Animal], [Couch, Furniture], [Happiness, Emotion]] } {CLASSES}, {PROPERTIES}

parameters based on previous work[45], and categorised the models as large above 30B parameters and small below (blue and orange respectively in Table 4).

For the open source models, we choose models from the Qwen family, Llama, Mistral and Deepseek. From the Qwen family, we selected Qwen2.5-Coder and Qwen 3. Qwen2.5-Coder is trained on 5.5 trillion tokens of code data[46]. It is optimised for code generation and reasoning, which might benefit our task. In contrast, Qwen3 is a general-purpose model trained with a Mixture-of-Expert (MoE) architecture[47]. From the Llama family, we choose Llama 3.3, a large general-purpose model from the Llama family[48], as it has been a prominent family of open source LLMs and proven competitive in performance on a variety of tasks[49]. Mistral small 3.1[50], also a general-purpose model, is included due to its strong performance-to-size ratio; Mistral models have been widely adopted due to efficient inference and high scores on both coding and general reasoning benchmarks at their size class[49]. Finally, the last open source model we include is Deepseek R1, a competitive open source reasoning model[51].

For the proprietary models, we focus on comparing general purpose models and reasoning models from OpenAI, along with their smaller variants. We include the general purpose models from the OpenAI GPT-4 family GPT-4o and GPT-4o-mini, and GPT-4.1 and GPT-4.1-mini. For the reasoning models we selected o1, o1-mini and o4-mini from the OpenAI-o-family. Previous studies have demonstrated that these models significantly outperform open source models in ontology learning tasks, such as term typing and identifying taxonomical relations[6,52]. More recently, Lippolis et al.[24] found that the o1 model yielded the most promising results, while the Llama-3.1 model exhibited the most structural flaws.

4.3 Evaluation

All results are automatically evaluated with precision, recall, and F1 scores. They are calculated by comparing the predicted axioms with the true axioms in the ontologies. For the disjoint axiom, symmetric

Table 4. Overview of selected language models (Params colored by size: Blue = Large, Orange = Small.

Model Name	Family	Params	Specialisation	Open	Exact Model
Qwen2.5-Coder	Qwen	14.8B	Reasoning / coding	Open	Qwen2.5-Coder-14B-Instruct
Qwen3	Qwen	30B	General purpose	Open	Qwen3-30B-A3B
Llama 3.3	Llama 3	70.6B	General purpose	Open	Llama-3.3-70B-Instruct
Mistral 3.1 Small	Mistral	23.6B	General purpose	Open	mistralai/Mistral-Small-3.1-24B-Instruct-2503
DeepSeek R1	Deepseek	32.8B	Reasoning / coding	Open	deepseek-r1:32b
GPT-4o	GPT4	∼200B	General purpose	Closed	GPT-4o-2024-08-06
GPT-4o-mini	GPT4	∼8B	General purpose	Closed	GPT-4o-mini-2024-07-18
GPT-4.1	GPT4	∼1.8T	General purpose	Closed	gpt-4.1-2025-04-14
GPT-4.1-mini	GPT4	∼8B	General purpose	Closed	gpt-4.1-mini-2025-04-14
o1	o	∼300B	Reasoning / coding	Closed	o1-2024-12-17
o1-mini	o	∼100B	Reasoning / coding	Closed	o1-mini-2024-09-12
o4-mini	o	∼100B	Reasoning / coding	Closed	o4-mini-2025-04-16

predictions are counted as correct, so A DISJOINT B is correct if the ontology states B DISJOINT A. Our experiments cover five axiom types, nine ontologies, two approaches, three shot settings, and twelve LLMs, yielding a total of 3,240 results. In the results section, we present analyses for each of these factors separately, taking the mean over the other factors.

Because LLMs do not consistently output the same results, we add a robustness evaluation, where we run our experiments three times.

5 Results

In this section, we present the results from our experiments described in the previous section. Section 5.1 presents the results on the different axioms. Section 5.2 zooms in on the results of the different ontologies. In Section 5.3, we present the results for the direct approach, the AbA (Axiom by Axiom) approach, and the shot settings. Section 5.3 presents the results over the two approaches and the three different shot settings. Section 5.4 presents the results for the different LLMs. Finally, we present the results of the robustness evaluation in Section 5.5.

5.1 Axioms

Table 5 shows the mean precision, recall, and F1 scores for the different axioms, averaged over the other parameters. The subclass axiom scores highest on all three scores, whereas the range axiom has the lowest F1 score, followed closely by the domain axiom. Precision scores are overall higher than recall, except for the domain.

5.2 Ontologies

Table 6 shows the results of identifying the different axioms per ontology. The ontology with the highest scores is the FOAF ontology, followed by the Pizza and gUFO ontology. The ERA ontology scores lowest. Figure 3 shows the F1 scores per axiom and ontology. Here it can also be observed that the FOAF ontology has the highest scores, although for the subproperty and disjoint axioms, gUFO has higher scores.

Table 5. Mean precision, recall, and f1 score per axiom.

Axiom	Precision	Recall	F1
disjoint	0.114	0.082	0.095
domain	0.034	0.043	0.038
range	0.050	0.021	0.030
subclass	**0.399**	**0.326**	**0.359**
subproperty	0.109	0.102	0.106

Table 6. Mean precision, recall, and f1 score per ontology.

Ontology	Precision	Recall	F1
ERA	0.049	0.063	0.055
FOAF	**0.257**	**0.194**	**0.221**
GoodRelations	0.152	0.098	0.119
gUFO	0.208	0.144	0.170
Music	0.146	0.098	0.117
NordStream	0.084	0.122	0.100
Pizza	0.215	0.143	0.172
SAREF	0.061	0.069	0.065
Time	0.099	0.103	0.101

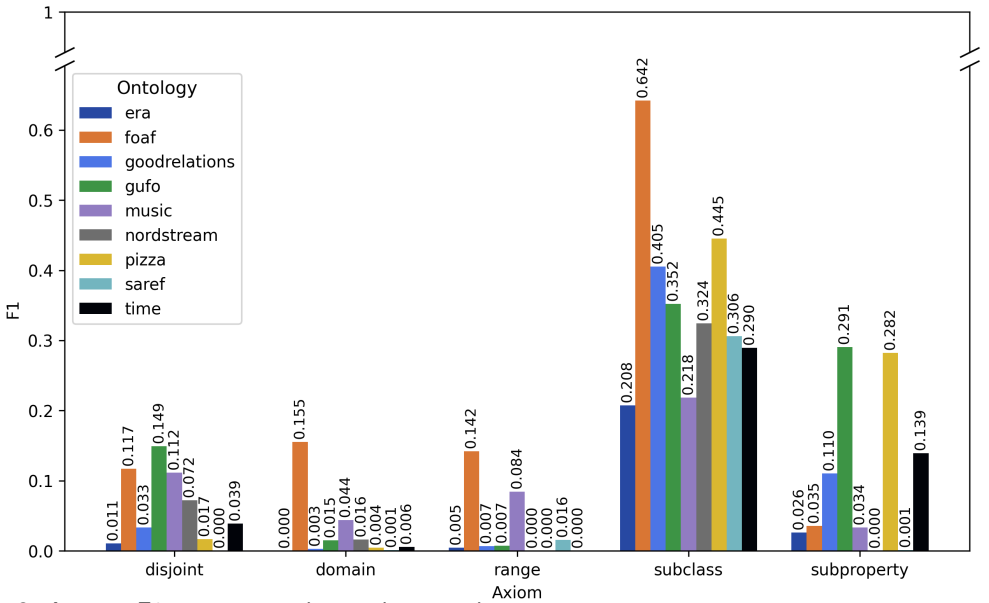

Figure 3. Average F1 scores per axiom and per ontology.

5.3 Approach & Shot settings

Table 7 shows the average F1 scores for the two different prompting approaches and the three shot settings. Overall, the AbA prompting method has the highest F1 score. On the shot level, we can observe the highest scores for the few-shot setting in the direct approach, and for the one-shot setting in the AbA approach. When looking at the different prompting approaches per axiom, shown in Figure 4, we can observe that AbA performs better for the subclass, domain, and range axioms. For the disjoint axiom, the direct approach works better, and for subproperty, the performance is almost equal.

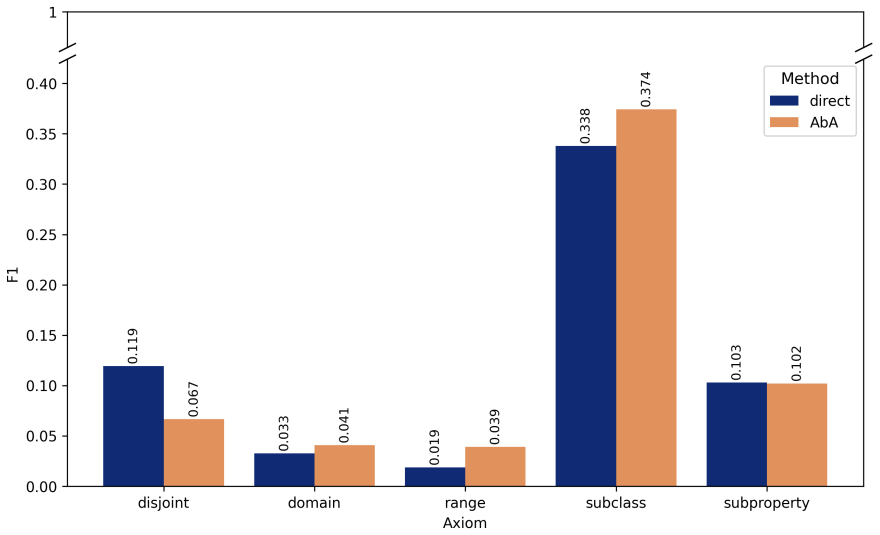

Figure 4. Average F1 scores per axiom and per prompting approach.

Table 7. Mean precision, recall, and F1 score per approach and shot setting.

Approach	Shot	Precision	Recall	F1
direct	few-shot	**0.159**	0.104	0.126
	one-shot	0.158	0.099	0.122
	zero-shot	0.158	0.100	0.122
AbA	few-shot	0.125	0.127	0.126
	one-shot	0.127	**0.133**	**0.130**
	zero-shot	0.120	0.128	0.124

Table 8. F1 scores by LLM category (GP = General Purpose, RC = Reasoning / Coding.

Closed	Open	Large	Small	GP	RC
0.124	0.077	**0.118**	0.077	0.101	**0.110**

Table 9. Mean precision, recall, and F1 score per LLM.

Model	Precision	Recall	F1
Qwen2.5-Coder	0.094	0.053	0.068
Qwen3	0.096	0.080	0.087
Llama 3.3	0.136	0.109	0.121
Mistral 3.1 Small	0.106	0.108	0.107
DeepSeek R1	0.130	0.060	0.083
GPT-4o	0.154	0.159	0.156
GPT-4o-mini	0.070	0.061	0.065
GPT-4.1	0.169	**0.185**	0.176
GPT-4.1-mini	0.137	0.126	0.131
o1	**0.249**	0.163	**0.197**
o1-mini	0.146	0.099	0.118
o4-mini	0.206	0.176	0.190

5.4 LLMs

Table 9 shows the mean precision, recall, and F1 score for each LLM. The o1 model scores highest overall (F1 0.164) due to its high precision of 0.249. GPT-4.1 has the highest recall (0.185), and the o4-mini has scores in the same range with an F1 score of 0.197. The models with the lowest scores are GPT-4o-mini, both Qwen models, and the Deepseek model.

Table 8 shows the performance of the models by their categories as presented in Section 4.2: Size, open source or proprietary (Open vs Closed), and specialisation (general purpose or reasoning / coding). Here,

it can be observed that the proprietary models by OpenAI perform better overall than the open source models. The large models outperform the small models, and the models specialised in reasoning / coding outperform the general purpose models, although the difference is small in this case (0.009).

5.5 Robustness

Table 10 shows the results of the three runs for an indication of how consistent the models perform. Overall, the differences are small, with the precision and recall having very minimal differences in run 1 and 2, and being a bit higher for run 3.

Table 10. Mean precision, recall, and f1 score per run.

Run	Precision	Recall	F1
run 1	0.139	0.113	0.125
run 2	0.139	0.114	0.125
run 3	**0.145**	**0.118**	**0.130**

6 Discussion

Axioms The results for different axioms indicate that subclass axioms are extracted most successfully, whereas domain and range axioms consistently perform worst. This can be partly explained by the fact that, semantically, words for subclass and superclass often lie closer together in meaning; therefore, it might be easier to recognise them. In contrast, difficulties with domain and range axioms might stem from context sensitivity. Unlike subclass axioms, their interpretation can vary depending on the specific context in which the ontology is applied, which likely contributes to the lower extraction performance.

Ontologies The domain of the ontology influenced how well axioms could be identified. The FOAF ontology stands out, achieving a score of 0.642 for the subclass axiom (Figure 3). In contrast, the ERA ontology gives notably lower results. These differences may stem from the domains: the FOAF domain is more general, making axioms easier to identify. Additionally, LLMs are likely trained on data that includes papers on this ontology, since it is an often-used ontology, further improving performance. On the other hand, the ERA ontology is more specialised, containing concepts such as SIDING and PHASEINFO, which may have less publicly available data.

Approach & Shot settings Prior work has demonstrated the influence of prompting methods on model performance[25]. In our experiments, we observe only a modest effect of prompting approach and shot settings. Few-shot and one-shot prompts produced slightly higher F1 scores in some cases, but the differences are limited. This may reflect the intrinsic complexity of axiom extraction: even providing examples does not substantially simplify the task, suggesting that model performance is more constrained by task difficulty than by prompt design.

LLMs Overall, the larger LLMs outperformed the smaller ones, with o1 achieving the best results. This is to be expected and in line with previous work; larger LLMs usually outperform the smaller ones, but are also more costly in energy consumption and computing time. Among smaller models, o4-mini performs surprisingly well, achieving a higher F1 score than all models except o1, although precise comparisons are difficult because OpenAI does not disclose model size details.

The proprietary models of OpenAI perform better than the open source models. However, they are more expensive to run, and technical details are not available. Within open source models, Llama 3.3 and Mistral Small perform best, occasionally outperforming some proprietary alternatives, highlighting that careful model selection can mitigate cost without sacrificing performance. Model specialisation had a smaller overall effect, though the two best-performing models (o1 and o4-mini) are reasoning models, suggesting that this task benefits from reasoning capabilities.

Finally, although LLMs can hallucinate and produce inconsistent results, our robustness test shows that, for this task, the differences are small and the general observations still hold.

Reflecting on this research, the results highlight the difficulty of this task, reflecting the observation that as we move up the ontology learning layer cake, automation becomes increasingly challenging. While LLMs have advanced the extraction of lower-layer knowledge, their effectiveness decreases for higher-layer tasks such as axiom identification. This is not just a limitation of LLMs: developing axioms is also challenging for humans. This suggests that a hybrid approach may be promising. For example, an LLM could generate candidate axioms, which a developer then evaluates and selects according to the specific use case. Alternatively, developers could provide additional documentation or requirements to guide the LLM in generating more relevant axioms. Experimenting with different types of input, such as domain-specific texts or competency questions, could further enhance this collaborative workflow.

6.1 Limitations

Even though we extensively compared different prompts, models, and ontologies, there are still aspects that could further improve results. First, we chose to focus on five RDFS/OWL axioms, but extending this work to additional axioms, complex axiom structures, and logical rules could be a valuable direction for future research. Second, we created a benchmark with nine medium-sized ontologies; it would be valuable to extend this benchmark with both more axiom types and ontologies.

In our evaluation, we have chosen a strict scoring method. However, for axioms such as SubClass and SubProperty, more lenient scoring would have been possible. E.g., if ‘a Pizza is a Food and a Food is a Thing’, then the prediction ‘a Pizza is a Thing’ could have been considered correct because of the transitive quality of these axioms. Additionally, we evaluated the results by taking averages over other parameters. When looking at combinations, such as the F1 scores per axiom and per ontology in Figure 3, more nuances become visible. We highlighted the combinations most relevant to our research goals, such as differences per axioms and per ontology, and axioms and prompting approach. However, more combinations are possible and potentially interesting.

Furthermore, while we spent effort on various settings, other prompting methods could be taken into account, such as Chain of Thought-prompting or varying the few-shot permutations. Finally, evaluating ontologies is inherently challenging, especially for automatically generated ones. Additional evaluation, such as with competency questions or having a human perform this task, could provide deeper insights into the quality and usefulness of the extracted axioms.

7 Conclusion

In this work, we explored how well LLMs can identify different types of axioms in ontologies. Formal ontologies and axioms are valuable for formalising knowledge within information systems, but their development is resource-intensive. To automate this process, we introduced a benchmark of nine

ontologies, 17.118 triples, and 2.771 axioms, and we tested multiple LLMs with different prompting approaches for axiom identification.

Our results indicate that some axiom types, like domain and range, are more challenging to identify than others, and the domain of the ontology significantly impacts performance. Additionally, prompting methods influence results, with the Axiom-by-Axiom prompting method being more successful overall than the Direct prompting approach. Furthermore, the larger LLMs consistently outperform smaller ones. Among those, o1 achieved the highest scores. Notably, for some ontologies and axiom types, F1 scores reached as high as 0.642, demonstrating that LLMs can achieve strong performance on this challenging task.

For future work, it would be interesting to extend this approach to additional axiom types and explore a more interactive workflow between LLMs and developers. This is especially worthwhile for identifying complex axioms, where ontology engineers remain essential. Overall, our findings suggest that LLMs have the potential to support the ontology engineering process. While fully automated axiom identification remains challenging, LLMs can provide valuable candidate axioms to support ontology engineers with the development and refinement of ontologies.

Acknowledgements

We would like to thank the TrustLLM project for their funding and support for this research. We are also grateful to Cornelis Bouter, who provided us with feedback on the axiom selection and labelled examples for the few-shot experiments.

References

1. Studer R, Benjamins VR and Fensel D. Knowledge engineering: Principles and methods. *Data & knowledge engineering* 1998; 25(1-2): 161–197.
2. Maedche A and Staab S. Ontology learning. In *Handbook on ontologies*. Springer, 2004. pp. 173–190.
3. Shamsfard M and Barforoush AA. The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review* 2003; 18(4): 293–316.
4. Buitelaar P, Cimiano P and Magnini B. *Ontology learning from text: methods, evaluation and applications*, volume 123. IOS press, 2005.
5. Asim MN, Wasim M, Khan MUG, Mahmood W and Abbasi HM. A survey of ontology learning techniques and applications. *Database, The Journal of Biological Databases and Curation* 2018; 2018: 1–24. DOI: 10.1093/database/bay101.
6. Babaei Giglou H, D'Souza J and Auer S. LLMs4OL: Large language models for ontology learning. In *International Semantic Web Conference*. Springer, pp. 408–427, 2023.
7. Lo A, Jiang AQ, Li W and Jamnik M. End-to-end ontology learning with large language models. *Advances in NIPS* 2024; 37: 87184–87225.
8. García-Fernández J, Verhoosel J, Ubacht J and Bakker RM. Ontology Engineering with Large Language Models: Unveiling the potential of human-LLM collaboration in the ontology extension process. In *Proceedings of the Fourth International Workshop on LLM-Text2KG at the Extended Semantic Web Conference (ESWC)*. Portorož, Slovenia: CEUR-WS.org, 2025.
9. Khadir AC, Aliane H and Guessoum A. Ontology learning: Grand tour and challenges. *Computer Science Review* 2021; 39: 100339.

10. Warren P, Mulholland P, Collins T and Motta E. The usability of description logics: understanding the cognitive difficulties presented by description logics. In *European Semantic Web Conference*. Springer, pp. 550–564, 2014.
11. Vigo M, Bail S, Jay C and Stevens R. Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. *International Journal of Human-Computer Studies* 2014; 72(12): 835–845. DOI: 10.1016/j.ijhcs.2014.07.005.
12. Corcho O and Gomez-Perez A. Evaluating knowledge representation and reasoning capabilities of ontology specification languages. In *ECAI'00 Workshop on Applications of Ontologies and Problem Solving Methods*. pp. 1–10, 2000.
13. Armary P, El-Vaigh CB, Labbani Narsis O and Nicolle C. Ontology learning towards expressiveness: A survey. *Computer Science Review* 2025; 56: 100693. DOI:10.1016/j.cosrev.2024.100693.
14. Giunchiglia F, Marchese M and Zaihrayeu I. Encoding classifications into lightweight ontologies. In *Journal on data semantics VIII*. Springer, 2007. pp. 57–81.
15. Fürst F and Trichet F. Heavyweight ontology engineering. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*. Springer, pp. 38–39, 2006.
16. Wong W, Liu W and Bennamoun M. Ontology learning from text: A look back and into the future. *ACM computing surveys (CSUR)* 2012; 44(4): 1–36.
17. Amalki A, Tatane K and Bouzit A. Deep Learning-Driven Ontology Learning: A Systematic Mapping Study. *Engineering, Technology & Applied Science Research* 2025; 15(1): 20085–20094. DOI:10.48084/etasr.9431.
18. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L and Polosukhin I. Attention is all you need. In *NIPS'17: Proceedings of the 31st International Conference on NIPS*. pp. 5998–6008, 2017.
19. Longpre S, Hou L, Vu T, Webson A, Chung HW, Tay Y, Zhou D, Le QV, Zoph B, Wei J et al. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*. PMLR, pp. 22631–22648, 2023.
20. Devlin J, Chang MW, Lee K and Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. pp. 4171–4186, 2019.
21. Wan Z, Cheng F, Mao Z, Liu Q, Song H, Li J and Kurohashi S. GPT-RE: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pp. 3534–3547. DOI: 10.18653/v1/2023.emnlp-main.214, 2023.
22. OpenAI. GPT-4 technical report, 2023.
23. Bakker RM, Di Scala DL and de Boer MHT. Ontology Learning from Text: an Analysis on LLM Performance. In *Proceedings of the 3rd NLP4KGC co-located with Semantics*. Amsterdam, Netherlands, pp. 17–19, 2024.
24. Lippolis AS, Saeedizade MJ, Keskisärkkä R, Zuppiroli S, Ceriani M, Gangemi A, Blomqvist E and Nuzzolese AG. Ontology generation using large language models. In *European Semantic Web Conference*. Springer, pp. 321–341, 2025.
25. Saeedizade MJ and Blomqvist E. Navigating Ontology Development with Large Language Models. In *The Semantic Web*, volume 14664. Springer Nature Switzerland, 2024. pp. 143–161. DOI:10.1007/978-3-031-60626-7_8. Series Title: Lecture Notes in Computer Science.
26. Ciatto G, Agiollo A, Magnini M and Omicini A. Large language models as oracles for instantiating ontologies with domain-specific knowledge. *Knowledge-Based Systems* 2025; 310: 112940.

27. Fathallah N, Staab S and Algergawy A. LLMs4Life: Large Language Models for Ontology Learning in Life Sciences. In *Proceedings of the EKAW 2024 Workshops, Tutorials, Posters and Demos, 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024)*. Amsterdam, The Netherlands, pp. 1–15, 2024.
28. Vaugrante L, Niepert M and Hagendorff T. A looming replication crisis in evaluating behavior in language models? evidence and solutions, 2024. ArXiv preprint, 2409.20303.
29. Wróblewska A, Podsiadły-Marczykowska T, Robert B, Protaziuk G and Rybinski H. Methods and tools for ontology building, learning and integration – application in the synat project. *Studies in Computational Intelligence* 2012; 390. DOI:10.1007/978-3-642-24809-2_9.
30. Haase P and Völker J. Ontology learning and reasoning—dealing with uncertainty and inconsistency. In *International Workshop on Uncertainty Reasoning for the Semantic Web*. Springer, pp. 366–384, 2005.
31. Bühmann L and Lehmann J. Pattern based knowledge base enrichment. In *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I 12*. Springer, pp. 33–48, 2013.
32. Ballout A, Tettamanzi AG and Pereira CDC. Predicting the score of atomic candidate owl class axioms. In *2022 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. IEEE, pp. 72–79, 2022.
33. Mateiu P and Groza A. Ontology engineering with Large Language Models. In *2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. pp. 226–229. DOI: 10.1109/SYNASC61333.2023.00038, 2023.
34. Gruber TR. A translation approach to portable ontology specifications. *Knowledge acquisition* 1993; 5(2): 199–220.
35. Rojas JA, Aguado M, Vasilopoulou P, Velitchkov I, Van Assche D, Colpaert P and Verborgh R. Leveraging Semantic Technologies for Digital Interoperability in the European Railway Domain. In *The Semantic Web – ISWC 2021*, volume 12922. Springer International Publishing. ISBN 978-3-030-88360-7 978-3-030-88361-4, 2021. pp. 648–664. DOI:10.1007/978-3-030-88361-4_38. Series Title: Lecture Notes in Computer Science.
36. Kalemi E and Martiri E. FOAF-Academic Ontology: A Vocabulary for the Academic Community. In *2011 Third International Conference on Intelligent Networking and Collaborative Systems*. pp. 440–445. DOI: 10.1109/INCoS.2011.94, 2011.
37. Hepp M. GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In *Knowledge Engineering: Practice and Patterns*, volume 5268. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-87695-3 978-3-540-87696-0, 2008. pp. 329–346. DOI:10.1007/978-3-540-87696-0_29. Series Title: Lecture Notes in Computer Science.
38. Almeida JPA, Guizzardi G, Sales TP and Falbo R. gUFO: a lightweight implementation of the Unified Foundational Ontology (UFO), 2019. URL http://purl.org/nemo/doc/gufo.
39. Yves R, Samer A, Mark S and Frederick G. The music ontology. In *Proceedings of the International Conference on Music Information Retrieval, ISMIR*, volume 2007. pp. 417–422, 2007.
40. Bakker RM and Di Scala DL. From text to knowledge graph: Comparing relation extraction methods in a practical context. In *First International Workshop on Generative Neuro-Symbolic AI, co-located with ESWC*, volume 4. p. 7, 2024.
41. Rector A, Drummond N, Horridge M, Rogers J, Knublauch H, Stevens R, Wang H and Wroe C. OWL Pizzas: Common errors & common patterns from practical experience of teaching OWL-DL. In *Proceedings of EKAW-2004*. Northampton, England: Springer Verlag, 2004.

42. García-Castro R, Lefrançois M, Poveda-Villalón M and Daniele L. The ETSI SAREF ontology for smart applications: a long path of development and evolution. *Energy Smart Appliances: Applications, Methodologies, and Challenges* 2023; : 183–215DOI:10.1002/9781119899457.ch7.
43. Pan F and Hobbs JR. Time ontology in owl. *W3C working draft, W3C* 2006; 1(1): 1.
44. Doimo D, Serra A, Ansuini A and Cazzaniga A. The representation landscape of few-shot learning and fine-tuning in large language models. *Advances in NIPS* 2025; 37: 18122–18165.
45. Abacha AB, Yim Ww, Fu Y, Sun Z, Yetisgen M, Xia F and Lin T. Medec: A benchmark for medical error detection and correction in clinical notes, 2024. ArXiv preprint, 2412.19260.
46. Hui B, Yang J, Cui Z, Yang J, Liu D, Zhang L, Liu T, Zhang J, Yu B, Dang K et al. Qwen2.5-coder technical report, 2024. ArXiv preprint, 2409.12186.
47. Qwen Team. Qwen3 technical report, 2025. 2505.09388.
48. Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, Letman A, Mathur A, Schelten A, Vaughan A, Yang A, Fan A, Goyal A, Hartshorn A, Yang A, Mitra A, Sravankumar A, Korenev A, Hinsvark A, Rao A et al. The llama 3 herd of models, 2024. ArXiv preprint, 2407.21783.
49. Minaee S, Mikolov T, Nikzad N, Chenaghlu M, Socher R, Amatriain X and Gao J. Large language models: A survey, 2024. ArXiv preprint, 2402.06196.
50. Mistral AI. Mistral small 3.1. https://mistral.ai/news/mistral-small-3-1, 2025.
51. DeepSeek-AI, Guo D, Yang D, Zhang H, Song J, Zhang R, Xu R, Zhu Q, Ma S, Wang P, Bi X, Zhang X, Yu X, Wu Y, Wu ZF, Gou Z, Shao Z, Li Z, Gao Z, Liu A et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. ArXiv preprint, 2501.12948.
52. Doumanas D, Soularidis A, Spiliotopoulos D, Vassilakis C and Kotis K. Fine-tuning large language models for ontology engineering: A comparative analysis of gpt-4 and mistral. *Applied Sciences* 2025; 15(4). DOI: 10.3390/app15042146. URL https://www.mdpi.com/2076-3417/15/4/2146.