Large Language Models as Assistants for Ontology Engineering

Mohammad Javad Saeedizade

Linköping University, Linköping, Sweden

Abstract

Ontology engineering is often a complex, time-consuming and costly process that relies heavily on expert engineers. Even experienced ontology engineers introduce errors, such as incompleteness in terms of requirements, and fall into common ontology pitfalls, underscoring the challenge of producing high-quality ontologies. This PhD proposal aims to address these issues by creating an LLM-based assistant for both ontology development and ontology evaluation. The envisioned assistant will offer suggestions during conceptual modelling, pattern-based suggestions for class and property definitions, and real-time validation checks to identify modelling errors. By embedding these capabilities into a unified tool, the research seeks to reduce dependence on expert intervention, enabling mid-level and novice ontology engineers and organisations to develop reliable ontologies more independently, while simultaneously accelerating the workflow of expert ontologists. The outcome of this work will be a software tool that supports and streamlines the ontology engineering lifecycle—facilitating creation, error detection, and quality assessment—thereby making ontology creation faster, less error-prone, and more accessible to non-experts.

Keywords

Large Language Models, Ontology Development, Ontology Evaluation, Ontology Engineering.

1. Problem statement

Ontology engineering is a challenging task that relies heavily on domain experts for both the creation and the evaluation of ontologies. Even when developed by ontology experts, ontologies frequently exhibit errors—ranging from pitfalls flagged by OOPS! (OntOlogy Pitfall Scanner!) [1], inaccuracy in the ontology, logical inconsistencies, completeness, and inadequate modelling of the intended requirements—that undermine their reliability and reuse. A key step in ensuring that an ontology meets its intended requirements is competency question (CQ) verification [2], in which an ontology engineer represents each CQ as a SPARQL query and evaluates it against the ontology. However, formulating and executing these queries imposes an additional burden on developers and further raises the barrier to entry for novice ontology engineers.

Meanwhile, many real-world tasks have been (semi-)automated through large language models (LLMs), such as GitHub Copilot [3]. The rapid pace of LLM development continually yields models that outperform their predecessors on a variety of knowledge-centric tasks. Although a handful of studies have explored the use of LLMs for discrete aspects of ontology engineering, no comprehensive tool currently exists to guide users through the end-to-end process of ontology modelling or evaluation. In this work, we propose tackling two tasks in ontology creation: (i) ontology development and (ii) ontology evaluation.

Based on eXtreme Design (XD), ontology creation is usually an incremental process of developing and evaluating [4]. Ontologists typically begin by framing a set of CQs and using ontology narratives as contextual background. They then represent the CQs and narrative fragments using OWL (the Web Ontology Language) [5] and propose a model. In the evaluation phase, they apply both structural and functional methods. Structural aspects are mostly measured with tools such as OntoMetric [6], consistency checking by running reasoners, OOPS! [1], FOOPS! [7], etc., which report structural statistics, logical issues, and common mistakes (pitfalls). On the other hand, functional methods

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan

javad.saeedizade@liu.se (M. J. Saeedizade)
0009-0000-0812-6167 (M. J. Saeedizade)

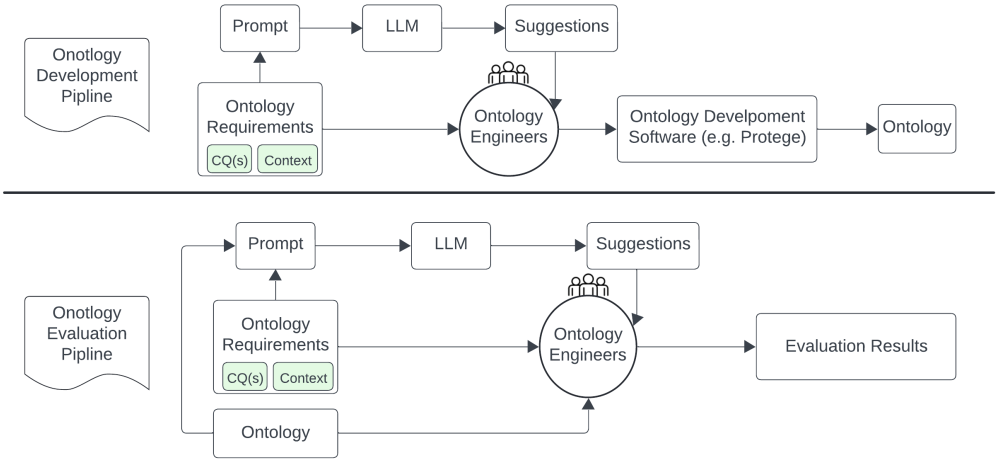


Figure 1: Two phases of ontology creation using an assistant tool. The top section illustrates ontology development, where requirements are used to generate suggestions that lead to the creation of an ontology. The bottom section depicts the evaluation phase, where the LLM is used to assist in the assessment of the ontology using the same set of requirements.

concern functionalities of the ontology, such as the intended modelling. These methods are mostly time-consuming and manual. One such method is CQ verification, which requires generating SPARQL queries for each CQ, executing them against the current ontology, interpreting the results to determine whether the ontology satisfies the intended requirement, and repairing any detected shortcomings in the next iteration of ontology revision.

In this work, we propose the development of an ontology engineering assistant that offers context-aware suggestions during conceptual modelling and semi-automates CQ verification by providing suggestions as shown in Figure 1. By embedding LLMs guidance and validation into a single pipeline, the tool is designed to accelerate ontology creation, reduce dependence on expert intervention, and make ontology engineering more accessible to novice ontology engineers.

2. Importance

The complexity of ontology engineering, from creation to maintenance, coupled with the necessity for expert knowledge engineers, often makes it challenging for organisations to use semantic web technologies and holds back novice ontology engineers, resulting in increased costs for organisations and creating an obstacle when it comes to adapting semantic web technologies.

By introducing an ontology engineering assistant that incorporates LLMs to create suggestions for ontologists during the development and evaluation phase, this work stands to benefit multiple stakeholders. Expert ontologists can streamline repetitive processes and reduce the risk of common errors, while novices can leverage guidance to accelerate their learning and contribute more effectively to ontology projects. Organisations gain the ability to develop cleaner, more maintainable ontologies with reduced expert involvement, thereby lowering overhead and fostering a wider adoption of semantic technologies. Ultimately, this research makes ontology engineering more accessible to companies by making this task simpler, faster and less costly.

3. Related work

LLMs for ontology development/generation. Recent studies have used LLMs to draft OWL ontologies from requirements. For example, Lippolis et al. [8] introduced Ontogenia, prompting LLMs to generate ontologies from user stories and CQs. In [9] and later in [10, 11], we likewise leverage LLMs to formalise requirements (CQs and user narratives), resulting in ontologies comparable to or better than those created by novice ontology engineers. Fathallah et al. [12, 13] developed NeOn-GPT and LLMs4Life pipelines for automated ontology modelling without user-based evaluation of the pipeline. In the work of Alharbi et al.[14], who developed DIAMOND-KG, we observed that participants' performance is significantly influenced by the accuracy of the LLM's predictions. Similarly, in our previous study[15], we demonstrated that more accurate knowledge graphs lead to improved performance in applications leveraging KGs. Mateiu & Groza [16] fine-tuned GPT-3 to translate natural language sentences into OWL axioms. They integrated their tool into Protégé, although they provided no proper evaluation of the tool. However, this tool is the closest thing to what this proposal is proposing.

LLMs for ontology evaluation. While the work mentioned in the previous paragraph has incorporated evaluation for their generated ontologies, evaluation has rarely been the central concern. For example, in Lippolis et al. [11] we proposed a set of criteria specifically designed for automatically generated ontologies; however, our framework still depended largely on manual judgments, revealing a broader need for more automated, scalable evaluation methods. Tsaneva et al. [17] used GPT-4's chat interface (ChatGPT) to compare automatically inferred axioms against human expert assessments, yet this work remained purely structural and was demonstrated on only a single toy ontology, limiting its broader applicability. Similarly, Benson et al. [18] examined GPT-4's capacity to both produce and critique class definitions within the Basic Formal Ontology. While they showed that a human-in-the-loop refinement process can enhance productivity in ontology tasks, their experiments were limited to a small set of illustrative classes and did not tackle functional evaluation aspects.

In our recent study [19], we investigated the use of LLMs to support ontology engineers in evaluating whether an ontology adequately addresses a given CQ. Our findings indicate that the performance of ontology engineers is strongly influenced by the accuracy of the suggestions provided by the LLM. Specifically, when the LLM offers a correct suggestion, the evaluators' performance improves substantially. Conversely, incorrect suggestions from the LLM lead to a marked decline in performance. It is important to note that, at this stage, we have not yet developed a tool suitable for deployment in industrial settings.

Thus, despite promising results, the use of LLMs for ontology evaluation, especially concerning functional adequacy, remains underdeveloped. In this context, Garijo et al. [20] leave the categorisation of existing resources on LLM use for ontology evaluation blank, highlighting this gap and pointing to directions where further research, with appropriate setup configurations, could yield more conclusive suggestions.

4. Research questions and hypotheses

This proposal to create an ontology creation assistant has two parts: (i) ontology development assistance (ii) ontology evaluation assistance. The main research questions related to the first component of the tool—ontology development assistance—are related to the capability of LLMs in creating ontologies by themselves or assisting ontology engineers in the development phase. They are as follows:

- **RQ1.1** To what extent can LLMs be used to support the generation of ontologies that meet a predefined set of requirements? Which LLMs and what prompting techniques are more effective?
- **RQ1.2** What evaluation criteria are suitable for evaluating LLM-generated ontologies?
- **RQ1.3** What are the strengths and weaknesses of ontologies generated using LLMs?
- **RQ1.4** To what extent can LLMs assist ontology engineers in ontology development, and what are the benefits and drawbacks of a hybrid approach combining LLM suggestions with expert validation compared to traditional human-only methods?

Similarly to the development phase, the following research questions examine the capabilities of LLMs in supporting ontology engineers during the evaluation phase:

- **RQ2.1** To what extent can LLMs evaluate ontologies using CQ verification?
- **RQ2.2** To what extent can LLMs assist ontology engineers in evaluating ontologies through CQ verification, and what are the benefits and drawbacks of a hybrid approach combining LLM suggestions with expert validation compared to traditional human-only methods?

This work hypothesises that LLMs can effectively assist ontology engineers in developing and evaluating ontologies.

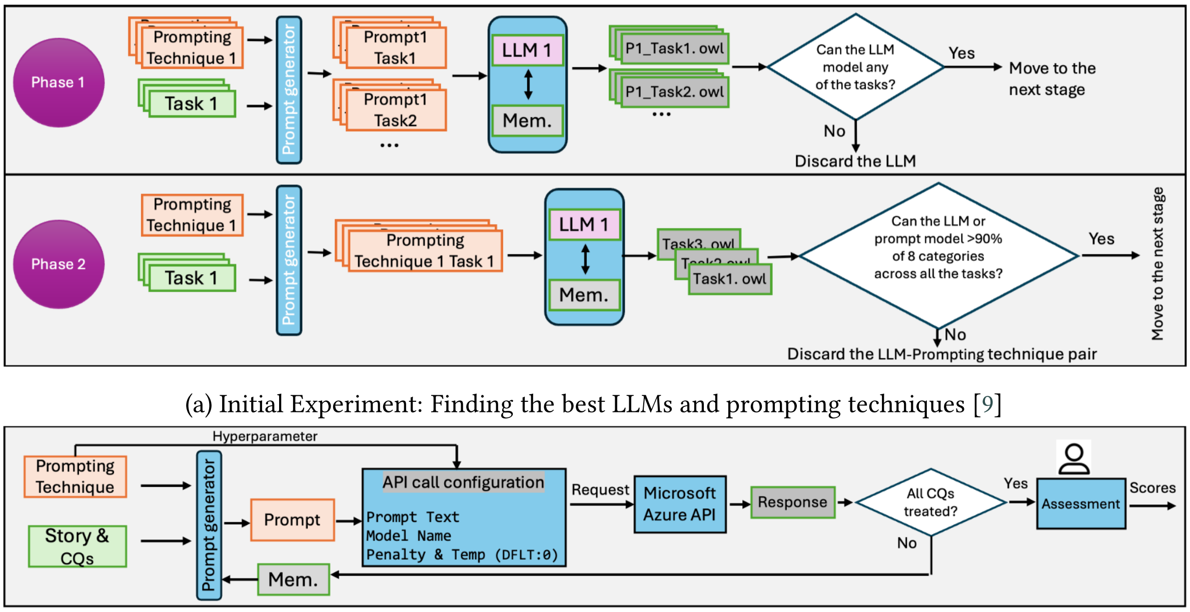
5. Preliminary results

There have been several experiments done to answer the research questions to some extent with reproducible results [21], and some work in the future work section will try to give more complete answers. The results are divided into two sections: (i) ontology development and (ii) ontology evaluation.

5.1. Ontology development

The research questions RQ1.1–RQ1.3 have been answered partially based on our previous work.

In our first work, Saeedizade and Blomqvist [9], we explored RQ1.1 through an automatic ontology generation pipeline shown in Figure 2. First, we filtered LLMs and prompting techniques on some simple ontology generation tasks shown in Figure 2a. Then, in the main experiment, Figure 2b, we evaluated the generated ontologies for the remaining LLMs and prompting techniques from the initial experiment. This work was submitted in December 2023, and at that time, GPT-4 and the sub-task decomposed prompting technique could generate ontologies similar to those of novice ontology engineers concerning the only presented criteria (based on CQ verification).



(a) Initial Experiment: Finding the best LLMs and prompting techniques [9]

(b) Main experiment, evaluating the generated ontologies manually.

Figure 2: The experiment setups of the first ontology generation work [9].

To extend this work and provide a better answer for RQ1.1 and also answer RQ1.2 and RQ1.3, we performed (To be presented in ESWC 2025) [11]. This work is shown in Figure 3, tackles several limitations of the previous work [9] by providing a holistic evaluation of the generated ontologies to address RQ1.2 and RQ1.3. Also, it shows that the generated ontologies are different from human-created ontologies; therefore, they need a different evaluation. We showed that some LLMs generated a lot of unnecessary classes and properties. Then we showed that counting unnecessary components, OOPS!,

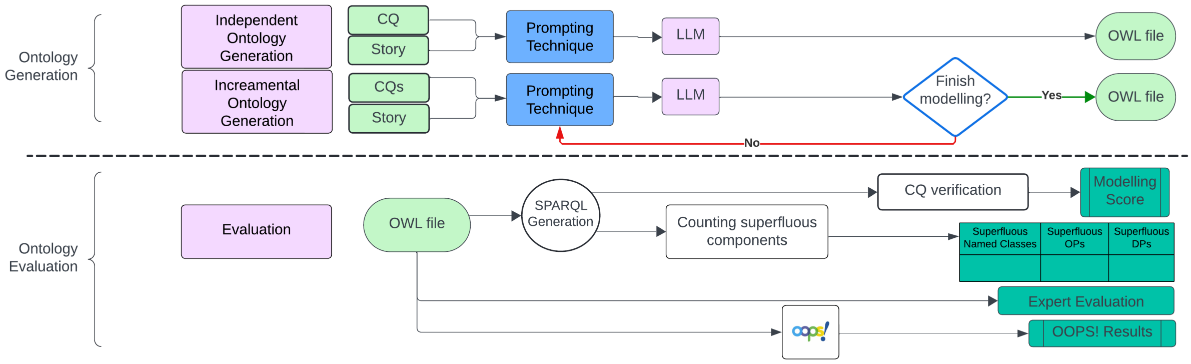


Figure 3: The Figure illustrates our second ontology generation work [11], which includes the generation setup and evaluation metrics to provide a holistic assessment

and CQ verification gives comparable evaluation results to expert ontology evaluation results. By December 2024, o1-preview with the introduced evaluation criteria outperformed novice ontology engineers and showed promising results to using the generated ontologies as a starting point for ontology development.

There were some risks related to the generalizability of LLMs on domain-specific tasks, which could result in LLMs performing well on one domain but poorly on another. We performed [10] (To be presented at ELMKE workshop at ESWC 2025) to investigate this risk, and we showed LLMs perform similarly on six domains used in the work. We also showed that some ontology development tasks that were considered complex, LLMs performed them with the same performance (concerning CQ verification) as simple ones.

5.2. Ontology Evaluation

We developed a prototype assistant for ontology evaluation, as described in our recent work [19] and illustrated in Figure 4. The primary objective of this assistant is to facilitate both automated and semi-automated verification of CQs.

In the upper part of Figure 4, a CQ, its context, and an ontology are provided to an LLM, which is then prompted to assess whether the ontology correctly models the given CQ. The LLM responds with a binary answer (yes or no), which is subsequently compared against a gold standard to evaluate performance. Using the `o1-preview` model, this automated approach achieved a macro-F1 score of 0.68.

The lower part of the figure illustrates the semi-automated evaluation process. In this setup, we presented the output generated by the LLM to ontology engineers, who were then asked to determine whether the ontology adequately models the CQ, using the LLM's suggestion as guidance with the ontology opened in Protégé. We deliberately mixed both correct and incorrect LLM suggestions across CQs that were either correctly or incorrectly modelled in the ontology. The results reveal that participants' performance was significantly affected by the accuracy of the LLM's predictions. Specifically, correct LLM suggestions improved human performance by 13%, while incorrect suggestions caused a decline of 26%. However, because the LLM provided more correct suggestions than incorrect ones, the opposing effects largely cancelled each other out, resulting in no net improvement in human performance.

In future work, we plan to extend this tool by incorporating visualisations and additional interactive features, guided by the feedback received during our experimental evaluation.

6. Evaluation

To evaluate the hypothesis in our chosen setup, the usefulness of LLMs in assisting ontology engineers in developing and evaluating ontologies should be assessed by means of a user-based study. We

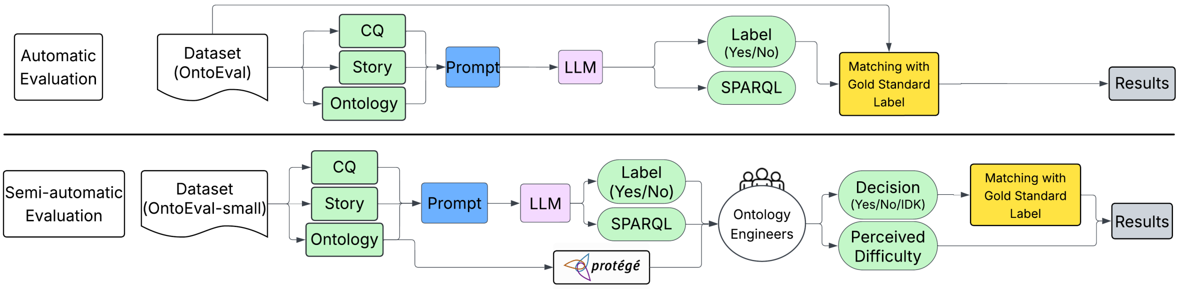


Figure 4: Overview of evaluation assistant: (i) Automatic: predicted labels are compared to gold standards to compute performance metrics. (ii) Semi-automatic: Users aided by LLM suggestions assess whether a CQ is modelled.

conducted some experiments to partially answer the mentioned research questions through a manual evaluation of LLMs' outputs by ontology engineers. Furthermore, we created a prototype to measure users' performance only in ontology evaluation with CQ verification. Overall, the main evaluation of tools for ontology engineering with LLMs as assistants should be done by measuring users' performance in ontology engineering, with and without LLMs' suggestions, and comparing their performance in each task.

After creating a tool for ontology engineering, we should hire/invite ontology engineers to develop and evaluate ontologies following our setup. To evaluate the users' performance using the tool, we will measure how accurately and efficiently ontology engineers perform the task of ontology engineering with the tool when (i) LLMs' suggestions are available to a user, and (ii) without LLMs' suggestions. The performance can be measured by measuring time to show how long it took users to complete the task, and assessing how accurate users were in each setting regarding ontology development and evaluation metrics in line with the selected CQ, such as measuring the proportion of correctly modelled CQs, quality of the final ontology concerning OOPS! warnings and expert evaluation.

7. Reflection and future work

In this paper, I propose creating an ontology engineering assistant tool. This tool helps ontologists to develop and evaluate ontologies by leveraging LLMs. Specifically, the approach relies on suggestions from LLMs to address relevant CQ throughout the ontology development process. For the ontology development part of the tool, the next step involves designing the appropriate setup and creating a tool, followed by defining a task that systematically measures its effectiveness via human-in-the-loop evaluation. Furthermore, during the ontology evaluation phase, we will capitalise on the user feedback we received to further enhance the tool by incorporating additional features and refined suggestions.

Declaration on Generative AI

The author declares the use of generative AI tools in preparing this manuscript. Specifically, ChatGPT were used for sentence polishing, paraphrasing, and grammar fixing; all AI-generated text was reviewed, edited, and approved by the author, who takes full responsibility for the final content.

Acknowledgments

I would like to express my gratitude to my supervisors, Eva Blomqvist and Robin Keskisärkkä, for their guidance, encouragement, and expertise throughout this work. Their insights and support have been instrumental in shaping both the direction and the quality of this work.

References

[1] M. Poveda-Villalón, A. Gómez-Pérez, M. C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation, International Journal on Semantic Web and Information Systems (IJSWIS) 10 (2014) 7–34.

[2] E. Blomqvist, A. Seil Sepour, V. Presutti, Ontology testing-methodology and tool, in: Knowledge Engineering and Knowledge Management: 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings 18, Springer, 2012, pp. 216–226.

[3] S. Peng, E. Kalliamvakou, P. Cihon, M. Demirer, The impact of ai on developer productivity: Evidence from github copilot, arXiv preprint arXiv:2302.06590 (2023).

[4] V. Presutti, E. Daga, A. Gangemi, E. Blomqvist, extreme design with content ontology design patterns, in: Proc. Workshop on Ontology Patterns, CEUR-WS, 2009, pp. 83–97.

[5] D. L. McGuinness, F. Van Harmelen, et al., Owl web ontology language overview, W3C recommendation 10 (2004) 2004.

[6] B. Lantow, Ontometrics: Putting metrics into use for ontology evaluation., in: KEOD, 2016, pp. 186–191.

[7] D. Garijo, O. Corcho, M. Poveda-Villalón, Foops!: An ontology pitfall scanner for the fair principles., in: ISWC (Posters/Demos/Industry), 2021.

[8] A. S. Lippolis, M. Ceriani, S. Zuppiroli, A. G. Nuzzolese, Ontogenia: Ontology generation with metacognitive prompting in large language models, in: European Semantic Web Conference, Springer, 2024, pp. 259–265.

[9] M. J. Saeedizade, E. Blomqvist, Navigating ontology development with large language models, in: A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, P. Lisena (Eds.), The Semantic Web, Springer Nature Switzerland, Cham, 2024, pp. 143–161.

[10] A. S. Lippolis, M. J. Saeedizade, R. Keskisarkka, A. Gangemi, E. Blomqvist, A. G. Nuzzolese, Assessing the capability of large language models for domain-specific ontology generation, arXiv preprint arXiv:2504.17402 (2025).

[11] 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, arXiv preprint arXiv:2503.05388 (2025).

[12] 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: European Semantic Web Conference, Springer, 2024, pp. 36–50.

[13] N. Fathallah, S. Staab, A. Algergawy, Llms4life: Large language models for ontology learning in life sciences, 2024.

[14] R. Alharbi, U. Ahmed, D. Dobriy, W. Łajewska, L. Menotti, M. J. Saeedizade, M. Dumontier, Exploring the role of generative ai in constructing knowledge graphs for drug indications with medical context, in: 15th International Semantic Web Applications and Tools for Healthcare and Life Sciences (SWAT4HCLS 2024) (to appear), 2024.

[15] M. J. Saeedizade, N. Torabian, B. Minaei-Bidgoli, Kgrefiner: Knowledge graph refinement for improving accuracy of translational link prediction methods, arXiv preprint arXiv:2106.14233 (2021).

[16] P. Mateiu, A. Groza, Ontology engineering with large language models, in: 2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, 2023, pp. 226–229.

[17] S. Tsaneva, S. Vasic, M. Sabou, Llm-driven ontology evaluation: Verifying ontology restrictions with chatgpt, The Semantic Web: ESWC Satellite Events 2024 (2024).

[18] C. Benson, A. Sculley, A. Liebers, J. Beverley, My ontologist: Evaluating bfo-based ai for definition support, arXiv preprint arXiv:2407.17657 (2024).

[19] A. S. Lippolis, M. J. Saeedizade, R. Keskisärkkä, A. Gangemi, E. Blomqvist, A. G. Nuzzolese, Large language models assisting ontology evaluation, 2025. URL: https://arxiv.org/abs/2507.14552. doi:10.48550/arXiv.2507.14552. arXiv:2507.14552, submitted July 19, 2025.

[20] D. Garijo, M. Poveda-Villalón, E. Amador-Domínguez, Z. Wang, R. García-Castro, O. Corcho, Llms for ontology engineering: A landscape of tasks and benchmarking challenges (2022).

[21] M. J. Saeedizade, R. Alharbi, H. B. Giglou, A. S. Lippolis, E. Blomqvist, V. Tamma, F. Grasso, T. R. Payne, J. D'Souza, S. Auer, et al., A framework for assessing llm consistency in knowledge engineering, 2025.