WIRTSCHAFTSUNIVERSITÄT WIEN
Vienna University of Economics and Business

Bachelor's Thesis

Title of Bachelor's Thesis (English)	Large Language Model usage when learning Ontology Engineering
Title of Bachelor's Thesis (German)	Nutzung von Large Language Modellen beim Erlernen der Ontologie-Entwicklung
Author (last name, first name):	Strubreiter Tobias
Student ID number:	12209863
Degree program:	Bachelor of Business and Economics, BSc (WU)
Examiner (degree, first name, last name):	Dipl.-Ing. Stefani Tsaneva, Prof. Dr. Marta Sabou

I hereby declare that:

1. I have written this Bachelor's thesis myself, independently and without the aid of unfair or unauthorized resources. Whenever content has been taken directly or indirectly from other sources, this has been indicated and the source referenced.
2. This Bachelor's Thesis has not been previously presented as an examination paper in this or any other form in Austria or abroad.
3. This Bachelor's Thesis is identical with the thesis assessed by the examiner.
4. (Only applicable if the thesis was written by more than one author): this Bachelor's thesis was written together with

The individual contributions of each writer as well as the co-written passages have been indicated.

27/02/2025
Date

Signature

Bachelor Thesis

Large Language Model usage when learning Ontology Engineering

Tobias Strubreiter

Date of Birth: 17.07.2002
Student ID: 12209863

Subject Area: Information Business

Studienkennzahl: J123456789

Supervisors:
Prof. Dr. Marta Sabou
Dipl.-Ing. Stefani Tsaneva

Date of Submission: 28.February 2025

Department of Information Systems & Operations Management, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

Contents

List of Figures

List of Tables

Abstract

Ontology engineering is a complex process that requires domain expertise, posing challenges for novice engineers. Large language models (LLMs) offer new possibilities for supporting ontology development, however their role and effectiveness remain underexplored. This thesis examines how novice engineers use LLM-based tools in ontology engineering, focusing on their impact on learning, workflow efficiency, and modeling accuracy. Through a mixed-methods approach, including a structured literature review, an online survey and follow-up semi structured interviews, we investigate how these tools assist in understanding ontology concepts and influence the overall engineering process. Findings highlight both benefits-such as automation and knowledge support-and challenges like contextual accuracy and reliance on LLM-generated suggestions. The study contributes to the evolving integration of artificial intelligence in ontology engineering, offering insights for improving LLM-assisted tools and workflows in this domain.

Acknowledgments

First of all, I would like to thank my supervisors, Prof. Dr. Marta Sabou and Dipl.-Ing. Stefani Tsaneva, for their continuous and helpful support during the creation of this thesis. Furthermore, I would like to thank my dear girlfriend, Tamara, my parents, Renate and Andreas, as well as my little sister, Emma, for their continuous support and encouragement during my bachelor's degree.

1 Introduction

The rapid evolution of advanced intelligent applications, particularly with the development of large language models (LLMs) such as OpenAI's ChatGPT and their versatile and modular capabilities have opened new paths when performing knowledge intensive tasks [31]. Recent advancements suggest that LLMs could play a key role in ontology engineering (OE) by providing computational support in areas such as knowledge elicitation, conceptual modeling and ontology learning (OL) [52, 13]. OE is a complex and systematic process to create, review and reuse structured representations of knowledge known as ontologies [52]. This process traditionally requires substantial manual effort, mainly in requirement collection and validation, leading to resource-intensive workflows [52]. To address this challenge, researchers have recently explored how LLMS can be leveraged to support these tasks [52, 13].

This thesis aims to explore the potential role of LLMs in OE. The study focuses on providing an overview of existing models that can support tasks within OE, while also investigating how they are utilized in practice by novice engineers. For the aforementioned goals the following research questions have been developed:

- How do novice ontology engineers make use of large language model supported tools when performing ontology engineering tasks? [RQ1]
- How do these tools assist novice engineers to understand the general domain of ontology engineering? [RQ2]
- How do these tools influence the overall approach to ontology engineering and in which parts of ontology engineering large language models are primarily used? [RQ3]

To achieve our objectives, we conducted a systematic literature review centered on current approaches and methodologies for using LLMs in OE. This thesis highlights their applications and potential to enhance various OE processes while also including studies that outline general approaches to OE. We analyzed these papers and organized already existing frameworks into a concept matrix, categorizing the domains within OE where LLM-supported frameworks can provide support. Furthermore, a survey was conducted among students, who attended an introductory course to ontology engineering, on how they used LLMs for their assigned OE tasks. From the survey participants, interview candidates were selected for follow-up discussions, and in-depth interviews were conducted.

As a result, this thesis serves as a valuable resource for novice ontology engineers and developers, guiding them in exploring the potential of LLMs to enhance and streamline their ontology development process.

The remaining of the thesis is structured as follows: In Sect. 2 we introduce Key terms and defnitions relevant for the thesis. After that Sect. 3 provides an overview of related literature, highlighting previous studies that explore the role of LLMs in OE. In Sect. 4, we outline the methodological approach of this study. Further in Sect. 5 we discuss in two Subsections how LLMs are relevant to OE. The first Subsect. 5.1 presents an overview of existing LLM-supported frameworks in ontology engineering. The other Subsect. 5.2 categorizes and analyzes different approaches to integrating LLMs into ontology development and also going over limitations and challenges. In Section 6, we analyze the results of the online questionnaire, focusing on how novice ontology engineers utilize LLMs, which tasks they apply them to, and their perceptions of effectiveness. Section 7 expands upon the questionnaire findings by presenting insights from semi-structured interviews. Finally, Section 8 synthesizes the key findings of this research, discussing the role of LLMs in OE, their benefits and limitations, and their impact on novice engineers.

2 Key Terms and Definitions

In this section the most important terms and concepts are presented that are necessary to fully understand the different approaches to OE and the tools and frameworks used for it.

Ontology. The word itself is taken from philosophy where it stands for a systematic explanation of Existence [17]. In the artificial intelligence (AI) field an ontology defines the basic terms, relations and rules for structuring, combining and extending the vocabulary of a topic field [17].

Knowledge Graphs (KGs) are structured representations of knowledge that interlink diverse entities and relationships to provide data integration and analysis [23]. Ontologies serve as a foundation for KGs by providing the conceptual schema to the governing domain, while assuring consistency, interoperability and coherent structure [23].

Ontology Engineering (OE) denotes the discipline that incorporates principles, methods and tools for creating and maintaining ontologies [21].

A plethora of OE methodologies have been introduced as of today which have either been proposed initially or emerged from experiences and insights gained during ontology development [21].

Human in the loop (HiL). A human-in-the-loop system in OE describes a collaborative process between LLMs and human experts in which LLMs generate initial ontology structures and humans iteratively refine those [8].

Natural language processing (NLP) is a branch of AI and linguistics which enables computers/machines to understand, interpret, generate and respond to human language [10].

Large Language Models (LLMs) are advanced AI systems designed to process and generate human-like text based on vast amount of collected data [40]. These models use Transformer architectures to handle complex NLP tasks [49, 40].

Prompt engineering describes a novel discipline which focuses on designing optimal instructions for LLMs to enable successful task performance [14]. Several main prompting techniques exist:

- *Zero-shot (Z-S) prompting.* In Z-S prompting a task is provided to an LLM without any examples, relying on its pre-trained knowledge to generate responses [43].
- *Few-shot (F-S) prompting.* For F-S prompting the model is given a few examples of input-output pairs to guide its generations of responses [43].
- *Chain-of-thought (CoT) prompting.* For CoT prompting the model is instructed to explicitly break down its reasoning process step by step improving its ability to tackle complex tasks [43].
- *Tree-of-thought (ToT) prompting.* ToT prompting is an extension to CoT prompting which allows the model to explore multiple reasoning branches in a structured tree-like path [43].
- *Role-play (R-P) prompting.* In R-P prompting the LLM is guided to act as a specific role or persona when performing a task which should help to generate a more context-decisive response [44].

- *Retrieval augmented generation (RAG).* RAG is a method that enhances a LLM's capability by dynamically integrating relevant external knowledge into the generation process which in turn enables more coherent and accurate responses [38, 43].
- *Meta-cognitive prompting (MP).* MP encourages the model to self-reflect and self-assess its response, enhancing the logical consistency and accuracy of the generated ontological structures [25].

3 Related Literature

The thesis emphasizes understanding how novice ontology engineers use LLMs to enhance their OE capabilities, improve their domain knowledge, and shift their overall approach to OE. Prior studies highlight how LLM-supported tools assist users at different stages of the OE process.

For example, in [3], the authors investigate how novice engineers use LLMs to construct models and perform tasks such as entity classification, relationship identification, and attribute specification. Participants were tasked with building conceptual models using LLM-support, and their processes were analyzed to determine where the model offered the most assistance. LLMs were highly effective in generating initial drafts and suggesting semantic relationships but users frequently needed to revise and refine the outputs to adhere to domain-specific requirements, complex dependencies, and contextual nuances. This revealed that while LLMs can streamline early-stage model creation, they require human input to maintain accuracy and consistence. The findings highlight the dual role of LLMs as both accelerators of model creation and scaffolding tools for learning conceptual modeling practice.

Similarly, in [42] the authors explore LLMs for developing ontologies and systematically evaluate their capabilities in different OE processes. In the study LLMs are used to generate OWL ontologies directly from ontology requirements. This study is one of the first to thoroughly evaluate the viability of LLM-supported OE.

In the OntoChat framework [52], the authors conducted a survey to identify which OE phases require the most support, finding that requirement elicitation, competency questions (CQs) extraction, and ontology testing were particularly challenging, especially for novices. OntoChat addresses these challenges by embedding LLMs into these tasks through a conversational interface that enables users to co-create user stories, refine CQs, and iteratively formalize requirements. The framework also clusters and verifies CQs to reduce redundancy and support ontology testing.

4 Methodology

To answer our research questions follow a mixed-methods approach. First, we conduct a systematic literature review, focusing on papers gathered mainly through Google Scholar. Second, we ask novice ontology engineers and developers to participate in an online Google Forms survey to gather first-hand knowledge of the current usage of LLMs by them in OE tasks. Last, we conduct follow-up interviews with some of the survey participants to gain more detailed insights into their approach to LLM-assisted ontology development.

4.1 Systematic Literature Review

For our systematic literature review (SLR) we adopted a topic-focused approach, emphasizing the exploration of research within a defined domain. This method synthesizes existing knowledge and identifies gaps related to a specific area of interest, as recommended in [22]. Our review focuses on the intersection of OE and LLMs. We conducted a *literature search* using combinations of these terms and related keywords in Google Scholar. In the end, we selected 22 publications based on their relevance, quality, and contribution to our research questions.

The *study selection* was performed by screening the title, release date, and abstract of each publication to ensure they aligned with our research objectives. A detailed overview of all selected publications and search strings can be found in Appendix A. To address RQ2 and RQ3 more effectively, we incorporated insights from literature on OE methodologies, domain-specific knowledge and already existing LLM-supported OE frameworks. The domain of LLMs in OE is emerging and rapidly evolving [31]. Consequently, the available literature is limited, with many contributions yet to undergo formal publication and predominantly hosted on platforms like arXiv. Furthermore, existing studies often do not explicitly address the needs and challenges faced by novice engineers, creating a gap that this SLR seeks to address.

4.2 Online Questionnaire and Semi Structured Interview

To answer RQ1 and RQ2 we set up an online questionnaire for novice engineers with follow up interviews. The questionnaire on the usage of LLMs for OE was open from December 11, 2024, to January 3, 2025 targeting students who had attended an introductory course to OE, titled “Semantic Artificial Intelligence Technologies for Knowledge Management”, held at the Vienna

University of Economics and Business. An invitation was sent by email to all students who completed this course and had an active student account.

The questionnaire consisted of 20 questions, structured as follows: 15 multiple-choice questions, 3 yes/no questions, 1 open question and 1 question to check for genuine answers. Among the 15 multiple-choice questions, 10 utilized a Likert scale which are commonly employed to evaluate perceptions, attitudes, and usability, rendering them suitable for this study [30]. They enable participants to express the level to which LLM-supported tools fulfilled their requirements in different areas of OE in a systematic and measurable way [30]. One question was included to ensure that participants read the questions thoroughly rather than skipping through them. Additionally, a screening question was placed at the end to determine whether participants were interested in taking part in a follow-up interview. The open-ended question was only presented to participants who indicated that they did not use any LLMs for OE. A blank version of the questionnaire can be found in Appendix B.

For further refinement of the insights gained from the questionnaire a total of three semi structured interviews (SSI) have been conducted. The selected interview partners are novice engineers who first completed the online questionnaire mentioned above and agreed to participate in further interviews. According to [29], semi structured interviews are ideal for exploring complex occurrences and phenomena in information systems research as they provide a flexible yet structured approach for data collection. This format allowed us to prepare key questions for our research questions while simultaneously retaining the possibility to collect unforeseen insights into the appliance of LLMs by novice engineers in OE. The participant received a one-pager introducing the topic of the thesis, along with the interview questions and a summary of their responses from the online Google Forms questionnaire. Each interview started with a short introductory talk before continuing with asking key questions stepwise. All interviews were voice recorded using OBS Studio [11]. The interview guideline and the one-pager, are provided in Appendix C.

5 LLM Relevancy to Ontology Engineering

To create a foundation on how novice ontology engineers can make use of LLMs when performing OE tasks, a systematic literature review was conducted to gain insights into the existing possibilities for leveraging LLMs in OE. Results are structured thematically to encompass the different subbranches of OE where LLMs can be of assistance. OE is not a strictly linear process; rather, its steps are deeply interwoven, allowing for iterative refinements and adaptations. LLMs contribute to multiple stages within ontology engineering, supporting tasks ranging from terminology extraction and ontology enrichment to semi-automatic knowledge graph construction [31, 26].

5.1 Overview of Existing LLM-supported Frameworks

This section provides an overview of existing frameworks that integrate LLMs, establishing a foundation for understanding their capabilities and applications.

OntoChat [52] is a conversational ontology engineering framework that makes use of LLMs to support requirement elicitation, competency question extraction, and ontology validation. By facilitating interactive user story creation and clustering competency questions through LLM-based paraphrase identification, it reduces ambiguities and enhances stakeholder collaboration. The framework also enables ontology testing without using SPARQL [19] by verbalizing ontologies into natural language for verification against extracted requirements.

NeOnGPT [13] demonstrates how structured methodologies like the NeOn framework [46] can improve LLM-assisted ontology generation. The NeOn-GPT framewok ensures that generated ontologies are logically sound, syntactically valid, and aligned with domain requirements by integrating LLMs within a structured ontology development pipeline. With NeOn-GPT the authors also mitigate some of the common limitations of LLM-generated ontologies, such as inconsistencies, lack of hierarchical depth, and missing axioms. In contrast to pure Z-S approaches, which often result in flat taxonomies and incomplete ontologies, NeOn-GPT incorporates prompt engineering techniques-including CoT , R-P and F-S prompting to guide LLMs through the three phases ontology specification, conceptualization, and implementation. In addition the NeOn-GPT pipeline is integrated with validation tools such as RDFLib [24] for syntax validation, HermiT reasoner [16] for logical consistency checking and OOPS! [39] for detecting and resolving modeling errors and pitfalls.

Ontogenia [25] is introduced as a new methodology that explores the use of LLMs in ontology generation, focusing on the application of MP to improve the design and refinement of ontologies. Ontogenia follows an iterative and incremental approach to ontology development, adhering to the eXtreme Design (XD) methodology, a structured framework in OE. Additionally, Ontology Design Patterns (ODPs) are integrated to facilitate knowledge structuring and transferability.

OntoKGen [1] is a LLM-driven system for automated ontology extraction and KG generation from technical documents within the reliability and maintainability (RAM) domain. The system incorporates LLMs through an adaptive iterative CoT algorithm ensuring structured ontology extraction and knowledge graph construction. OntoKGen integrates Neo4j [50] , a non-relational database, for flexible knowledge storage and retrieval.

LLMs4OL [14] is a conceptual framework that allows us to leverage LLMs through zero-shot OL for term typing, taxonomy discovery, and non-taxonomic relation extraction. The authors tested in their approach eight major types of domain-independent LLMs across multiple open-source ontological knowledge sources: WordNet, GeoNames, UMLS and schema.org. Among all tested models, GPT-4 [35] outperformed the others across all three before mentioned OL tasks, achieving the highest accuracy in term typing, taxonomy discovery, and non-taxonomic relation extraction. The study found that larger models generally performed better, and while domain-specific models like PubMedBERT were tested, which did not consistently outperform general-purpose LLMs. Furthermore, the study revealed that fine-tuning LLMs such as Flan-T5 significantly improved their performance, making them competitive with GPT-4 despite having fewer parameters.

LLMs4OM is a framework with novel approach designed to enhance OM using LLMs [15]. OM is crucial for knowledge integration, as it aligns heterogeneous ontologies to facilitate data interoperability and knowledge sharing. LLMs4OM introduces a RAG pipeline, leveraging Z-S prompting across three ontology representations: concept, concept-parent, and concept-children. The framework operates through two main modules: a retrieval module based on RAG that selects candidate matches from a knowledge base and a matching module that refines selections using LLM-based similarity assessments.

OLaLa , presented in [20], is a ontology matching system using LLMs to improve knowledge graph alignment. The study explores the integration of Z-S and F-S prompting techniques with open-source LLMs to enhance entity resolution. OLaLa incorporates a structured matching pipeline, utilizing pretrained language models, retrieval strategies, and prompt engineering to refine ontology alignment.

DRAGON-AI gets introduced by the authors of [47] as a dynamic RAG system designed to facilitate ontology construction and maintenance using LLMs. The system automates ontology term completion, generating textual definitions and logical relationships by leveraging knowledge from existing ontologies and unstructured sources such as GitHub issues.

5.2 LLM-supported Ontology Engineering

In this section we provide an overview of OE activities that LLMs can potentially assist and discuss which of the frameworks, presented in 5.1 support each OE activity.

For a better overview, Table 1 provides a mapping between the frameworks and the OE tasks. We discuss each of these OE tasks and how they have been approached with LLM-based systems in Sect.5.2.1-5.2.5.

Table 1: LLM-supported OE frameworks and their assistance to specific OE tasks. *Abbreviations:* RE = Requirement Elicitation, OD = Ontology Development, OL = Ontology Learning, OP = Ontology Population, OA = Ontology Alignment, OM = Ontology Matching.

OE Task **Framework**	**RE** **(S.6.1)**	**OD** **(S.6.2)**	**OL** **(S.6.3)**	**OP** **(S.6.4)**	**OA and OM** **(S.6.5)**
OntoChat [52]	X	X	-	-	-
NeOnGPT [13]	X	X	X	X	-
Ontogenia [25]	X	X	X	-	-
OntoKGen [1]	X	X	X	-	-
LLMs4OL [14]	-	X	X	-	-
LLMs4OM [15]	-	-	-	-	X
OLala [20]	-	-	-	-	X
DRAGON-AI [47]	X	X	X	X	X

5.2.1 Requirements Elicitation using LLMs

Requirements elicitation describes the process of extracting and structuring knowledge from human experts for the purpose of building knowledge-based systems, ontologies and KGs [23]. For this typically CQs are used as means of determining scope of an ontology and ensuring the ontology contains necessary knowledge to answer domain-specific queries [33].

Application Idea Generation. LLM-supported application idea generation in OE primarily enhances creativity, automation, and domain alignment by generating structured concepts, suggesting ontology-based applications, and refining ontology requirements iteratively [25, 52, 13, 1]. There are several different methodologies on how LLMs support application idea generation. OntoChat supplements idea generation by enabling interactive requirement elicitation through dialogue-based CQs extraction, allowing users to iteratively refine and validate potential ontology applications [52]. Ontogenia on the other hand leverages metacognitive prompting MP to guide LLMs in self-reflecting on generated concepts, leading to more structured and contextually relevant ontology-based application ideas [25]. NeOnGPT provides a structured pipeline for domain-specific ontology modeling by translating natural language descriptions into formal ontologies, allowing users to explore application ideas through ontology-driven structured prompts [13]. OntoKGen integrates RAG to enhance domain knowledge retrieval, supporting users in iteratively refining ontology-based application ideas while ensuring coherence with structured knowledge graphs [1].

Scope Definition and Competency Questions Generation. CQs play a crucial role when defining ontology scope and ensuring functionality [33]. In [52] the authors propose the framework OntoChat which enables a conversational workflow where users can interactively extract and refine CQs with additional support through clustering and verification of CQs as well as removal of redundant ones. A similar semi-automatic process is also suggested in [23] where the authors propose using LLMs to generate CQs through a semi-automatic pipeline before ontology construction.

A further human-in-the-loop approach was taken in [38] which centers around automating CQ generation in OE by leveraging LLMs in combination with retrieved domain-specific knowledge. The process involves domain knowledge indexing where relevant papers are converted into vector embeddings and stored in a retrieval system. Afterwards relevant data is retrieved from the system where top-k similar text chunks based on cosine similarity are fetched to enrich the LLMs contextual understanding. As a last step a response is generated where the retrieved knowledge serves as a context for generating more accurate and domain-specific competency questions.

A more automated approach is taken by NeOn-GPT which employs the NeOn methodology [46] to provide an automated pipeline for ontology creation [13]. Phase one is focused on specifying ontology requirements encompassing domain description, purpose, scope, ontology requirements and example CQs. By making use of structured prompts and techniques such as CoT and R-P prompting, GPT-3.5 [51] is guided to produce domain-relevant CQs that define the scope and requirements of the ontology.

LLMs can also be used to retrofit CQs [2]. The RETROFIT-CQ approach suggested in [2] uses LLMs, particularly open-source models, to generate CQs based on ontology triples, reversing the usual process where CQs define ontology content. Moreover, Ontogenia further enhances CQ generation by employing Metacognitive Prompting (MP), a novel prompting technique aimed at improving self-assessment and logical consistency in LLM-generated CQs where the LLM actively self-reflects on its reasoning process during ontology design.

5.2.2 LLM-supported Ontology Development

Ontology development is the process of formally defining concepts, relationships, and constraints within a specific domain to facilitate knowledge representation and sharing [33]. LLMs can assist in many tasks required for ontology creation notable requirement specification, ontology conceptualisation and ontology implementation [13]. One key advantage of LLMs is their ability to generate domain-specific concepts and relationships by analyzing vast amounts of textual data making expert-level insights over different domains more accessible [8]. This helps novice ontology engineers identify relevant entities, attributes, and hierarchical structures more efficiently [42]. Furthermore, LLMs can improve knowledge representation by converting unstructured or semi-structured data into formal ontology components, making integration with semantic web technologies smoother [14, 23]. A major part in leveraging LLMs to their full extent is prompt engineering, which determines the accuracy and consistency of ontology extraction [4, 42].

Specifically, fine-tuning GPT-3 [9] to convert textual descriptions into OWLFS has proven to be successful in automating important ontology engineering tasks. This method ensures a formal and logically sound ontology structure by making it easier to define class subsumption, domain and range constraints, object property relationships, disjoint classes, and cardinality limits [26]. Fine-tuning is only one part of a broader LLM-supported OE landscape [42]. Furthermore the authors explore various LLM architectures, comparing fine-tuned models, zero-shot learning with GPT-4, and structured prompting techniques to assess the quality of OWL [6] ontologies generated from textual descriptions highlighting through a multi-phase experiment that GPT-4 consistently outperformed GPT-3.5 and smaller open-source models in generating OWL [6] ontologies [42]. The authors also underscore that hybrid approaches combining LLMs with human oversight and error correction yield the best results in real-world scenarios [42]. For example, a modular OE approach is proposed in [45] where LLMs can be used for taxonomy extraction and relation identification, reducing the manual workload required for defining class hierarchies and non-taxonomic relationships. This method improves efficiency by allowing LLMs to propose structural patterns, which are then refined through human intervention [45]. A similar approach introduces the X-HCOME methodology, which integrates human expertise with LLM-assisted ontology development [8]. This framework enhances the quality of generated ontologies by iteratively refining class definitions and relations through human-LLM interaction, ensuring both completeness and consistency [8].

5.2.3 Ontology Learning using LLMs

OL focuses on automatically acquiring and structuring knowledge from textual sources to build ontologies [14]. Traditional ontology creation is manually intensive, requiring domain experts, but OL techniques aim to automate this process by identifying terms, types, relations, and axioms using AI applications and data mining [14]. Traditional OL approaches are based on lexico-syntactic pattern mining and clustering [14]. Recent developments in NLP using LLMs provide us with a suitable alternative to traditional OL methods [14]. The main goal of OL is to generate a cost-effective and scalable solution for knowledge acquisition [14].

The LLMs4OL conceptual framework allows us to leverage LLMs through zero-shot OL for term typing, taxonomy discovery, and non-taxonomic relation extraction. The study found that larger models generally performed better, and while domain-specific models were tested, they did not consistently outperform general-purpose LLMs [14]. Furthermore, the study revealed that fine-tuning smaller LLMs significantly improved their performance, making them competitive with larger models despite having fewer parameters [14]. LLMs can serve as powerful tools to assist ontology engineers, particularly when fine-tuned for OL tasks [14]. Another approach would be that in phase two of the NeOn methodology [46], where ontology conceptualisation is the main goal, GPT-3.5 is utilised through F-S prompting the CQs generated in phase one help to extract key entities, relationships, and properties, which are then structured into subject-relation-object triples [13]. Out of the generated triples, GPT-3.5 is prompted to create a full ontology depcited in Turtle syntax [7] in phase three, ontology implementation [13].

Another approach is taken in [1] where the authors focus on extracting ontologies and knowledge graphs directly from targeted knowledge from users. They introduce OntoKGen which revolves around adaptive iterative CoT prompting which iteratively identifies, validates, and refines concepts, relationships, and attributes to create a structured ontology by first collecting targeted knowledge from users-either predefined ontologies or specific text-and then using structured steps and user interaction to refine ontology extraction [1]. A different but related approach was taken by the authors in their work on DRAGON-AI [47]. While OntoKGen primarily uses an adaptive CoT algorithm for ontology extraction, DRAGON-AI employs a RAG methodology that integrates LLMs with structured and unstructured knowledge sources to dynamically generate ontology components to complete ontologies [1, 47].

5.2.4 Ontology Population using LLMs

Ontology population (OP) is the process of adding instances to an ontology, enriching it with real-world data to make it applicable for practical use. This process involves extracting, structuring, and integrating data to fit within the predefined ontology schema [32].

LLMs can be powerful tools for OP, particularly in extracting structured knowledge from unstructured text [32, 45]. The approach demonstrated in [32] shows that LLMs can succesfully extract 90% of correct triples when guided by modular ontologies, while also relying on RAG and text summarization to bypass the token limit when creating context information. In [45] the authors propose segmenting an ontology into modular components to enhance LLM performance in OP, particularly by improving conceptual consistency and maintaining a focused scope, meaning smaller and more concise prompts, for knowledge extraction. Both papers suggest that LLMs, when paired with a modular ontology as guidance in prompts, can significantly automate ontology population while still requiring human validation to ensure precision [32, 45].

5.2.5 Ontology Alignment and Matching using LLMs

Ontology alignment (OA) describes the process of establishing correspondences between concepts, relations, and instances in different ontologies to make them interoperable [15]. Ontology matching (OM), a subtask of OA, focuses on automatically detecting equivalent entities across ontologies and is particularly well-suited for LLMs due to their ability to process and interpret natural language descriptions, relations, and structures [20]. This involves identifying similar or equivalent entities across two or more ontologies and map them accordingly [45].

LLMs4OM introduce a dual-module strategy which consists of retrieval module based on RAG and a matching module optimized for zero-shot prompting across three ontology representations (concept, concept-parent, and concept-children) [15]. In addition to integrating RAG techniques for canditate selection a further approach uses RAG and text summarization techniques to enhance entity linking and minimize hallucinations as well as evaluating different string similarity metrics to improve semantic alignment between extracted entities and ontology concepts effectively tackling some of the issues mentioned in LLMs4OM [15, 32]. The framework LLMs4OM was tested on 20 OM datasets from different domains with seven LLMs and four different retrieval techniques [15]. Similarly OLaLa suggests using prompting techniques (Z-S and F-S) with multiple LLMs to explore OM tasks [20]. A minor difference lies in candidate selection, while OLaLa solely relies on sentence-BERT models to identify potential correspondences, LLMs4OM employs multiple retrieval models, including sentence-BERT to generate candidate matches [20, 15]. After candidate selection, both systems implement additional filtering mechanisms, including consistency and cardinality checks, ensuring the final alignments maintain high accuracy [20, 15]. Both OLaLa and LLMs4OM show that LLMs can outperform traditional OM methods with the correct implementation of retrieval models, prompting strategies and filtering mechanisms [20, 15].

5.2.6 Challenges and Limitations of LLMs in OE

LLMs face several challenges in OE, particulary the lack of logical consistency, ambiguity and inconsistent responses [31]. LLMs struggle to integrate formal semantics, often reducing ontologies to textual descriptions while missing the logical frameworks that define them [37]. Entity disambiguation remains a critical challenge since LLMs tend to fuse similar terms together, which is especially troubling in OA [37]. Additionally they fail to enforce logical consistency, generating conflicting axioms which require human intervention to correct them [31, 26]. Scalability and query accuracy also remains a problem because LLMs struggle with handling large and complex ontologies and often generate queries which are syntactically correct but amount to no returns [28]. Furthermore LLM-driven ontology evaluation reveals that while models like GPT-4 achieve high accuracy in detecting certain errors, they still struggle with subtle logical inconsistencies and require multiple representations of the same ontology to improve verification accuracy [48]. LLMs also struggle with adapting to new domains, because the new domain lacks sufficient structured training data, making ontology generation and refinement highly dependent on human oversight [41, 28]. A major limitation is that LLMs lack real-world grounding, making them vulnerable to hallucinations and generating information that does not align with verified knowledge bases [31, 1, 13]. Efforts to integrate RAG have shown promise in reducing hallucinations, but they still rely on the quality of retrieved data, which remains inconsistent [28, 32].

6 LLM Usage by Novice Ontology Engineers: Survey Results

In this section the findings from the online questionnaire are taken into account to gain intermediate insights and possible answers to RQ1 and RQ2. Nineteen students (N=19) participated in the questionnaire, with 94.7% (18 out of 19) stating that they used LLMs for learning ontology engineering. The remaining 5.3%, which corresponds to a single student, did not provide a clear explanation for their choice of not using any LLMs, simply stating:"At the moment I am not using any LLMs.".

6.1 Preferred Models and Application of LLMs in OE

OpenAI's ChatGPT [34] turns out to be the preferred model (100%) among the LLMs used by the students as seen in Fig. 1. Other models such as Google's Gemini [12] and Microsoft's Copilot [27] were used to a lesser extent (16.7% and 11.1% respectively), while Anthropic's Claude [5] was not used at all.

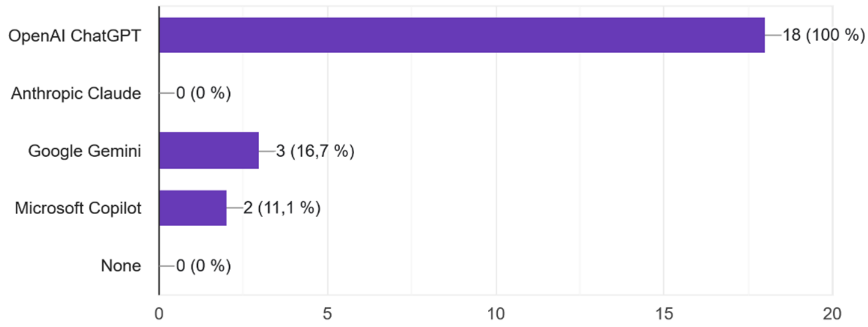

Figure 1: Question 3: Which widely known popular models did you use for Ontology Engineering?

In further questions, respondents were asked whether they had heard of LLM-supported frameworks for OE. The results, shown in Fig. 2, indicate that familiarity with these tools is relatively low. OntoChat was the most recognized, with 17.6% of respondents indicating that they had heard of it. DeepOnto was known to 11.8%, while NeOn-GPT and LLMs4OL were each mentioned by 5.9% of participants. However, a significant number of respondents (58.8%) stated that they had not heard of any of these frameworks and also did not name any other frameworks which were not listed in the questionnaire.

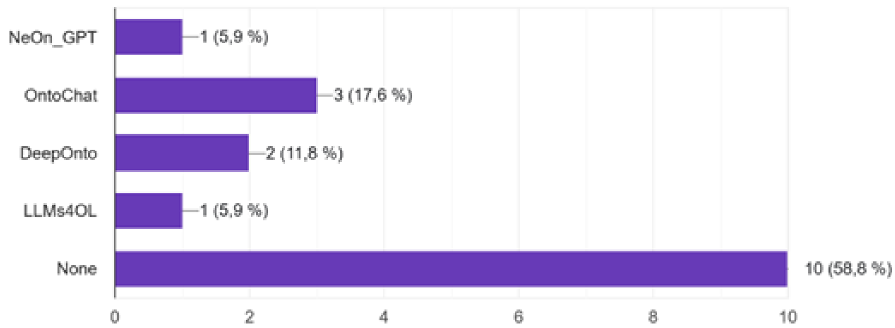

Figure 2: Question 5: Which LLM-supported frameworks for Ontology Engineering have you heard of?

Participants were also asked about the specific tasks within OE they had attempted to complete using LLMs. The results in Fig. 3 indicate that LLMs have been applied across various stages of the ontology development process, though with varying levels of adoption. Query generation emerged as the most common use case, with 61.1% of respondents employing LLMs for tasks such as generating SPARQL queries. Ontology population, which involves the creation of instances, was also a major use case, reported by 55.6% of participants. Ontology creation, specifically the generation of classes and relations, was attempted by 50% of respondents, further highlighting the role of LLMs in automating structural aspects of ontology engineering.

Ontology improvement, including the detection of inconsistencies and refinement of ontologies, was reported by 38.9% of respondents, the same percentage as those who attempted competency question generation. Advanced ontology definition, which involves generating property characteristics and constraints, was attempted by 33.3% of participants. The least common applications were requirements elicitation for ontology-based application idea generation and reasoning (inferring logical statements), each attempted by 27.8% of respondents.

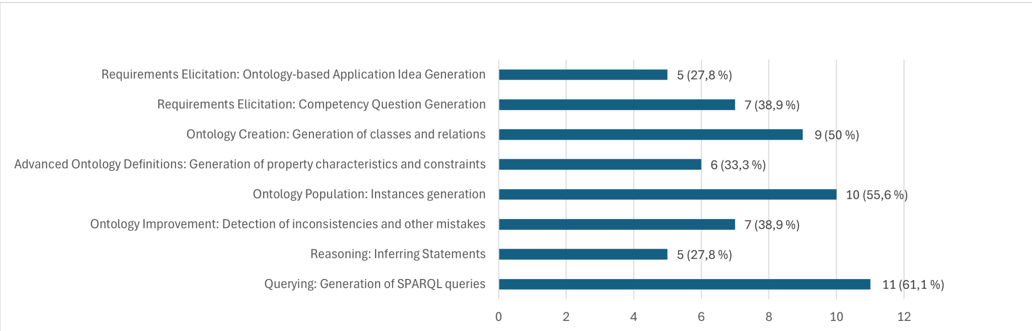

Figure 3: Question 6: What specific tasks within Ontology Engineering have you attempted to complete using LLMs?

6.2 User-Induced Experience of Effectiveness of LLMs in OE

Further respondents evaluated the effectiveness of LLMs across different OE tasks, revealing mixed results as shown in Fig. 4.

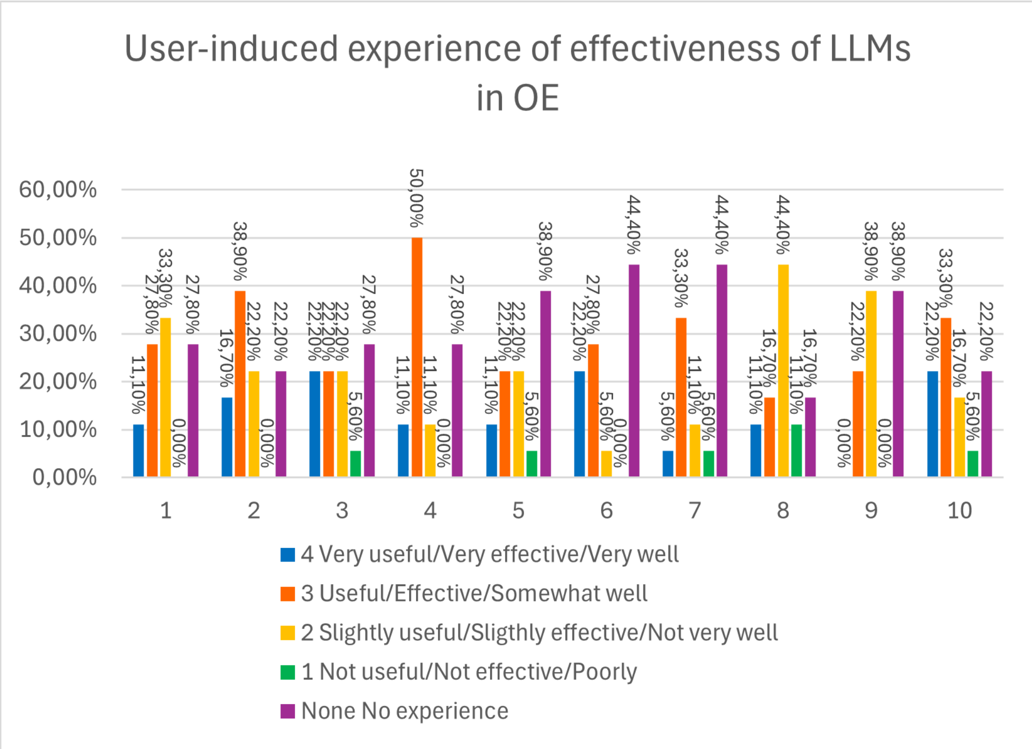

Figure 4: User-Induced experience of effectiveness of LLMs in the following OE tasks: (1) application idea generation, (2) CQ generation (3) Defining properties and relationships, (4) accurate class hierarchies suggestion, (5) generating property characteristics and ontology restrictions, (6) instance generation, (7) extracting terms from unstructured text, (8) validate consistency or logical soundness, (9) reasoning tasks and interference of additional triples, (10) SPARQL query generation

For a more effective evaluation the mean of each question has been computed by assigning numerical values to each Likert Scale category following the approach outlined by Nemoto and Beglar [30]. Each response was mapped to a scale ranging from 4 (Very Effective/Very Useful/Very Well) to 1 (Not Effective/Not Useful/Poorly), while responses marked as "No Experience" were excluded from the mean calculation to prevent them from skewing the results [18]. To compute the mean for each question, the aggregated sum of all valid responses was divided by the number of people contributing to the Likert scale (Total number of participants who used LLMs - Participants who marked "No experience"). This ensures that the computed mean accurately reflects the perceived effectiveness of LLMs across various OE tasks.

Table 2: Overview of computed means per question

Question	Mean	Calculation
1	2.69	35/13
2	2.93	41/14
3	2.85	37/13
4	3.00	39/13
5	2.64	29/11
6	3.30	33/10
7	2.70	27/10
8	2.33	35/15
9	2.36	26/11
10	2.93	41/14

The evaluation of LLM effectiveness across various ontology engineering tasks yielded mixed but generally positive results as shown in Table 2. The highest-rated aspect was the usefulness of LLMs in creating instances for ontologies (Mean = 3.30), followed by their effectiveness in suggesting accurate class hierarchies (Mean = 3.00) and generating SPARQL queries (Mean = 2.93). Competency question generation (Mean = 2.93) and defining properties and relationships (Mean = 2.85) also received relatively high scores. Conversely, tasks such as validating ontology consistency (Mean = 2.33) and reasoning-based inference (Mean = 2.36) were rated lower, indicating that participants found LLMs less effective in these areas.

6.3 Challenges and Error Handling

Respondents were asked about the general challenges they faced when generating ontology concepts using LLMs. The most frequently reported issue as shown in Fig. 5 were problems with logical consistency in generated concepts, affecting 66.7% of the participants. This suggests similar to the literature that LLMs often struggle to maintain coherence when defining ontology elements, leading to inconsistencies in structure or meaning [31]. 38.9% of the respondents encountered a lack of precision in domain-specific knowledge, indicating that LLMs do not always produce accurate or contextually relevant ontology concepts. 22.2% of the participants reported difficulties in managing context-specific terms, outlining another limitation in how well LLMs can handle specialized terminologies within different domains. Furthermore, 16.7% of respondents selected “All of the above”, suggesting that some users faced multiple challenges simultaneously. Meanwhile, 11.1% of the participants indicated that they had no experience with these challenges, implying that they either did not attempt to generate the concept of the ontology with LLM or did not encounter significant issues.

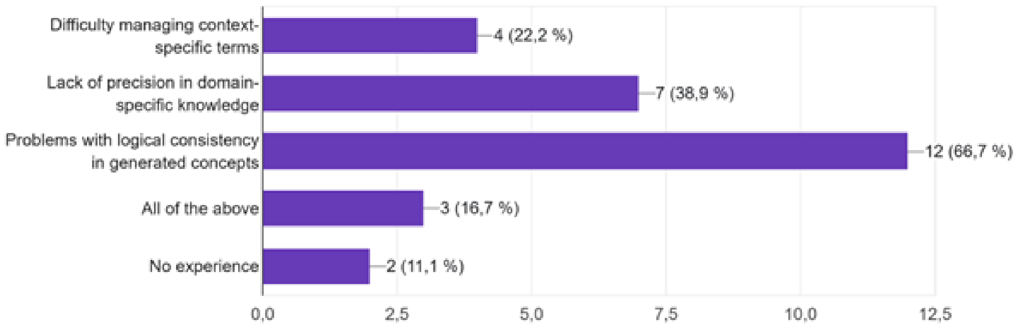

Figure 5: Question 19: What general challenges did you face when generating ontology concepts using LLMs?

In addition, respondents were also asked how they handled incorrect or incomplete outputs generated by LLMs in OE tasks. The results are shown in Fig. 6. The most common approach, reported by 66.7% of participants, was to manually correct the outputs, indicating that human oversight remains essential when working with LLM-generated content. Additionally, 55.6% of respondents combined LLM suggestions with human input, suggesting that while LLMs can provide useful starting points, their outputs often require human refinement. Another 44.4% of participants used iterative prompting or query refinement, highlighting an effort to improve LLM-generated results by adjusting input queries. Meanwhile, 11.1% of respondents avoided using LLMs for critical ontology tasks, possibly due to reliability concerns. Lastly, 5.6% of respondents reported having no experience dealing with incorrect or incomplete LLM outputs.

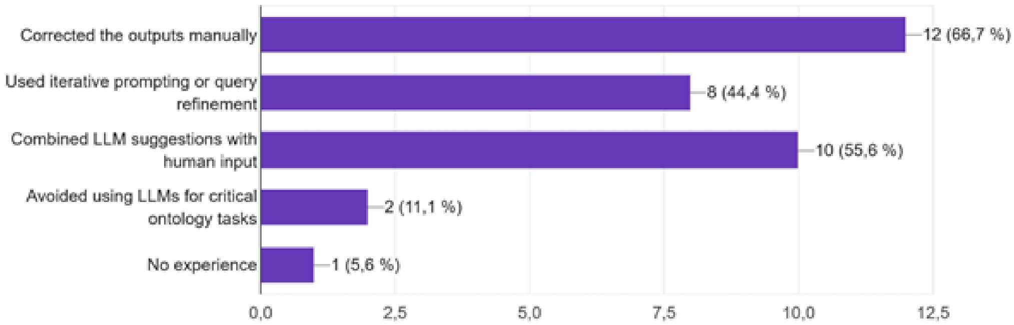

Figure 6: Question 20: How did you handle incorrect or incomplete outputs generated by LLMs?

7 LLM Usage by Novice Ontology Engineers: Interviews Insights

In the following section, we present the outcomes of the SSIs, which offer deeper insights into the intermediate insights for RQ1 and RQ2 generated by the online questionnaire. Out of the nineteen participants, three agreed to participate in follow-up interviews. All of them chose to remain anonymous.

7.1 Background and Experience of the Participants

All three participants are novice ontology engineers. Person B and C study business, economics and social sciences with a major in business informatics and Person A studies business law.

7.2 Usage of LLMs when Learning OE

All participants highlighted that their experience in OE was mainly gathered through university coursework. In detail one participant described their experience of learning OE as initially frustrating but gradually improving and more understandable through practical application and term explanation using LLMs. Two of them also stated that they used LLMs to explain important terms to them and provide example ontologies and explanations to them. One of them emphasized to structure an ontology in a way that the ontology adds logical value to the domain. While this was initially difficult, their ability to do so evolved throughout the ontology development process. Another one echoed this, stating that while structuring and designing ontologies was difficult at the start, practical exercises with LLM-provided explanations helped them to understand the logic behind class hierarchies and relationships.

7.2.1 Practice Examples

All three participants reported using general-purpose LLMs such as ChatGPT (3.5 , 4 and 4o [36]) and Google Gemini [12] rather than any specialized LLM supported OE frameworks. Two of them stated that they used LLMs for both technical and conceptual tasks with a focus on generating classes, relations and properties. identifying errors in ontology structures, and OP instances generation. One participant used ChatGPT primarily to generate instances for ontology population and assist with SPARQL query creation. All three noted that prompt engineering played a crucial role in obtaining useful results while two fo them also explicitly said that the generated elements often required iterative human refinements to correct errors and inconsistencies especially with older models like GPT 3.5.

7.2.2 Motivational Factors

Convenience, reduction of workload and term explanation were named as the primary motivators for using LLMs in their ontology work. Person A noted that ChatGPT acted as some sort of substitute for an information technology professional, significantly reducing the complexity of technical tasks. All three participants appreciated that ChatGPT provided structured and simple explanations for OE terms and processes, making it easier to understand domain concepts without extensive prior domain knowledge. They also all state that LLMs significantly reduced their workload in OE tasks.

7.2.3 Limitations

Several limitations were encountered by the interview participants. One participant pointed out that ChatGPT generated ontology components in incompatible formats which needed to corrected through iterative prompting and manual work. They also noted that ChatGPT 3.5 required well-structured prompts for effective use, whereas newer versions have improved usability. Two participants both mentioned that complex prompts often led to errors due to token limitations, requiring them to break up prompts into smaller parts and iteratively prompt the tasks. All three agreed that human intervention was essential to correct inaccuracies and refine LLM-generated outputs. None of them mentioned of having encountered hallucinations of LLMs.

7.3 Process Understanding and Efficiency

All participants reported that using LLMs significantly improved their efficiency in performing ontology-related tasks. One of them explained that ChatGPT's ability to provide real-time explanations helped them understand ontology structuring faster in comparison with traditional course materials. They all noted that learning by doing, supported by AI-generated explanations, facilitated a deeper understanding of ontology concepts. One participant stated that ChatGPT was highly effective in generating examples and assisting with ontology population, reducing the mental workload required to complete tasks. Another one agreed, stating that LLMs provided structured explanations and hands-on support, making it easier to grasp class hierarchies, properties, and constraints. Regarding external validation, one of the participants mentioned that they occasionally consulted the instructor for clarification, particularly when ontology population outputs were unclear. However, they found that LLM-generated explanations were generally more comprehensible than course slides. The other two participants did not rely on expert validation, as they found LLM outputs sufficiently accurate for their needs.

7.4 Reflections

When asked about potential improvements to LLM-based tools for ontology engineering, all three participants agreed that better integration with publically available LLMs would be beneficial, like a ChatGPT plugin. One participant suggested developing frameworks similar to GitHub Copilot for OE, where LLMs could be directly embedded into ontology development environments. All participants emphasized the importance of prompt engineering and recommended that prompt engineering techniques are tailored especially for OE. Further all of them stated that the major barrier to adoption of specialized ontology frameworks is their lack of accessibility and ease of use compared to general-purpose LLMs. They argued that if specialized tools or frameworks were as intuitive as ChatGPT or Gemini, they might see wider adoption among students and novice engineers.

8 Conclusion

This thesis aimed to investigate the role of LLMs in OE by providing an overview of how LLMs can support OE tasks and examining how novice engineers utilize them in practice. The study was guided by the following research questions:

- How do novice ontology engineers make use of large language model-supported tools when performing ontology engineering tasks? [RQ1]
- How do these tools assist novice engineers in understanding the general domain of ontology engineering? [RQ2]
- How do these tools influence the overall approach to ontology engineering, and in which parts of ontology engineering are large language models primarily used? [RQ3]

To address these questions, we employed a systematic literature review to examine the current state of LLM applications in OE. The review categorized existing LLM-supported frameworks and identified their relevance to various OE tasks. Furthermore, we conducted an online survey among students enrolled in an introductory OE course, investigating their use of LLMs in completing OE assignments. Follow-up semi-structured interviews provided deeper insights into their experiences and perceptions of LLM-supported tools.

The findings from the literature review, online survey, and semi-structured interviews collectively provide insights into the current role of LLM-supported tools in OE, particularly for novice engineers. The literature review highlights the growing interest in integrating LLMs into OE tasks, with various frameworks demonstrating the potential of LLMs in tasks like requirement elicitation, ontology development, and ontology learning [52, 13, 14]. However, major limitations persist, including logical inconsistencies, hallucinations, and the difficulty of enforcing domain-specific constraints [31, 37, 28].

Empirical findings from the questionnaire confirm that LLMs, particularly general-purpose models like ChatGPT, are widely adopted by novice engineers. Despite the existence of specialized frameworks, most respondents (58.8%) were unaware of them, indicating that accessibility and ease of use are critical adoption factors. The survey results suggest that students primarily use LLMs for query generation (61.1%), ontology population (55.6%), and class hierarchy structuring (50%), aligning with findings from studies identified through the SLR, especially in [42, 14]. However, LLM effectiveness varies across tasks with the highest ratings for ontology population and class hierarchy generation, while reasoning and logical validation were rated lower. These findings align with prior research emphasizing LLMs' struggles with logical coherence and verification [48].

Insights from the semi-structured interviews further support these trends, as all three participants relied on ChatGPT for both conceptual and technical OE tasks, favoring convenience and term explanations over domain-specific frameworks. Prompt engineering played a crucial role in optimizing responses, consistent with findings from the SLR [4, 42]. While LLMs significantly reduced workload and improved efficiency, the need for iterative corrections and manual refinement was emphasized. Interestingly, none of the interviewees explicitly reported hallucinations, which contradicts some concerns raised in literature [31, 1], possibly indicating that novice users primarily rely on LLMs for well-defined, lower-risk tasks rather than critical ontology development stages.

Overall, LLMs have shown promise in reducing entry barriers for novice ontology engineers by streamlining repetitive tasks, providing structured explanations, and supporting iterative development. However, their effectiveness is task-dependent, and without further advancements in logical consistency, ontology alignment, and domain specialization, they will continue to require human oversight [26]. In conclusion we gave an extensive overview of the many use cases of LLMs in OE and gave insights how novice ontology engineers make use of them. Future research should focus on improving the integration of LLM-supported OE tools into publicly available LLMs, making them more intuitive and readily available.

Declaration of generative AI and AI-assisted technologies in the writing process: ChatGPT was used for grammar checks, synonym suggestions, and assistance with LaTeX syntax, particularly for table generation and figure configuration.

References

[1] Mohammad Sadeq Abolhasani and Rong Pan. Leveraging llm for automated ontology extraction and knowledge graph generation, 2024.

[2] Reham Alharbi, Valentina Tamma, Floriana Grasso, and Terry Payne. Investigating open source llms to retrofit competency questions in ontology engineering. *Proceedings of the AAAI Symposium Series*, 4:188–198, 11 2024.

[3] Syed Juned Ali, Iris Reinhartz-Berger, and Dominik Bork. How are llms used for conceptual modeling? an exploratory study on interaction behavior and user perception. In Wolfgang Maass, Hyoil Han, Hasan Yasar, and Nick Multari, editors, *Conceptual Modeling*, pages 257–275, Cham, 2025. Springer Nature Switzerland.

[4] Bradley P Allen, Lise Stork, and Paul Groth. Knowledge engineering using large language models. *arXiv preprint arXiv:2310.00637*, 2023.

[5] Anthropic. Claude: A conversational ai model. Anthropic AI Model, 2023.

[6] Grigoris Antoniou and Frank van Harmelen. *Web Ontology Language: OWL*, pages 91–110. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[7] David Beckett, Tim Berners-Lee, Eric Prudâ€™hommeaux, and Gavin Carothers. Rdf 1.1 turtle. *World Wide Web Consortium*, pages 18–31, 2014.

[8] Georgios Bouchouras, Pavlos Bitilis, Konstantinos Kotis, and George A. Vouros. Llms for the engineering of a parkinson disease monitoring and alerting ontology. volume 3749, 2024. Cited by: 0.

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[10] Abhimanyu Chopra, Abhinav Prashar, and Chandresh Sain. Natural language processing. *International journal of technology enhancements and emerging engineering research*, 1(4):131–134, 2013.

[11] OBS Studio Contributors. Obs studio: Open broadcaster software, 2023. Version 29.1.3.

[12] Google DeepMind. Gemini: Multimodal ai model. Google DeepMind AI Model, 2023.

[13] Nadeen Fathallah, Arunav Das, Giorgis De Giorgis, Andrea Poltronieri, Peter Haase, and Liubov Kovriguina. Neon-gpt: A large language model-powered pipeline for ontology learning. In *Special Track on Large Language Models for Knowledge Engineering, Extended Semantic Web Conference, 2024. (ESWC 2024)*, 2024.

[14] Hamed Babaei Giglou, Jennifer D'Souza, and Soeren Auer. Llms4ol: Large language models for ontology learning, 2023.

[15] Hamed Babaei Giglou, Jennifer D'Souza, Felix Engel, and Sören Auer. Llms4om: Matching ontologies with large language models. *arXiv preprint arXiv:2404.10317*, 2024.

[16] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. Hermit: An owl 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269, October 2014.

[17] Asunción Gómez-Pérez. Ontological engineering: A state of the art. *Expert Update: Knowledge Based Systems and Applied Artificial Intelligence*, 2(3):33–43, 1999.

[18] Spencer E. Harpe. How to analyze likert and other rating scale data. *Currents in Pharmacy Teaching and Learning*, 7(6):836–850, 2015.

[19] Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language. World Wide Web Consortium (W3C) Recommendation, March 2013.

[20] Sven Hertling and Heiko Paulheim. Olala: Ontology matching with large language models. In *Proceedings of the 12th Knowledge Capture*

Conference 2023, K-CAP '23, page 131â€"139, New York, NY, USA, 2023. Association for Computing Machinery.

[21] Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha, Nurfadhlina Mohd Sharef, et al. An analysis of ontology engineering methodologies: A literature review. *Research journal of applied sciences, engineering and technology*, 6(16):2993–3000, 2013.

[22] Barbara Ann Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 07 2007.

[23] Vamsi Krishna Kommineni, Birgitta König-Ries, and Sheeba Samuel. From human experts to machines: An llm supported approach to ontology and knowledge graph construction. *arXiv preprint arXiv:2403.08345*, 2024.

[24] Daniel Krech, Gunnar AAstrand Grimnes, Graham Higgins, JÃ¶rn Hees, Iwan Aucamp, Niklas LindstrÃ¶m, Natanael Arndt, Ashley Sommer, Edmond Chuc, Ivan Herman, Alex Nelson, Jamie McCusker, Tom Gillespie, Thomas Kluyver, Florian Ludwig, Pierre-Antoine Champin, Mark Watts, Urs Holzer, Ed Summers, Whit Morriss, Donny Winston, Drew Perttula, Filip Kovacevic, Remi Chateauneu, Harold Solbrig, Benjamin Cogrel, and Veyndan Stuart. Rdflib, August 2023.

[25] Anna Sofia Lippolis, Miguel Ceriani, Sara Zuppiroli, and Andrea Giovanni Nuzzolese. Ontogenia: Ontology generation with metacognitive prompting in large language models. In *Proc. of the Extended Semantic Web Conference-Satellite Events, Springer, Crete, Grece*, 2024.

[26] Patricia Mateiu and Adrian Groza. Ontology engineering with large language models. In *2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 226–229. IEEE, 2023.

[27] Microsoft. Copilot: Ai-powered assistance. Microsoft AI Model, 2023.

[28] Ozan Baris Mulayim, Lazlo Paul, Marco Pritoni, Anand Krishnan Prakash, Malavikha Sudarshan, and Gabe Fierro. Large language models for the creation and use of semantic ontologies in buildings: Requirements and challenges. In *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 312–317, 2024.

[29] Michael D Myers and Michael Newman. The qualitative interview in is research: Examining the craft. *Information and organization*, 17(1):2–26, 2007.

[30] Tomoko Nemoto and David Beglar. Likert-scale questionnaires. In *JALT 2013 conference proceedings*, volume 108, pages 1–6, 2014.

[31] Fabian Neuhaus. Ontologies in the era of large language models – a perspective. *Applied Ontology*, 18(4):399–407, 2023.

[32] Sanaz Saki Norouzi, Adrita Barua, Antrea Christou, Nikita Gautam, Andrew Eells, Pascal Hitzler, and Cogan Shimizu. Ontology population using llms, 2024.

[33] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[34] OpenAI. Generative pre-trained transformer (gpt). OpenAI Model, 2023.

[35] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[36] OpenAI. Gpt-4o system card. OpenAI, 2024.

[37] Jeff Z Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, et al. Large language models and knowledge graphs: Opportunities and challenges. *arXiv preprint arXiv:2308.06374*, 2023.

[38] Xueli Pan, Jacco van Ossenbruggen, Victor de Boer, and Zhisheng Huang. A rag approach for generating competency questions in ontology engineering. *arXiv preprint arXiv:2409.08820*, 2024.

[39] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):7–34, 2014.

[40] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[41] Francesco Ronzano and Jay Nanavati. Towards ontology-enhanced representation learning for large language models. *arXiv preprint arXiv:2405.20527*, 2024.

[42] Mohammad Javad Saeedizade and Eva Blomqvist. Navigating ontology development with large language models. In Albert Meroño Peñuela, Anastasia Dimou, Raphaël Troncy, Olaf Hartig, Maribel Acosta, Mehwish Alam, Heiko Paulheim, and Pasquale Lisena, editors, *The Semantic Web*, pages 143–161, Cham, 2024. Springer Nature Switzerland.

[43] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

[44] Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. *Nature*, 623(7987):493–498, 2023.

[45] Cogan Shimizu and Pascal Hitzler. Accelerating knowledge graph and ontology engineering with large language models. *arXiv preprint arXiv:2411.09601*, 2024.

[46] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. *The NeOn Methodology for Ontology Engineering*, pages 9–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[47] Sabrina Toro, Anna V Anagnostopoulos, Susan M Bello, Kai Blumberg, Rhiannon Cameron, Leigh Carmody, Alexander D Diehl, Damion M Dooley, William D Duncan, Petra Fey, et al. Dynamic retrieval augmented generation of ontologies using artificial intelligence (dragon-ai). *Journal of Biomedical Semantics*, 15(1):19, 2024.

[48] Stefani Tsaneva, Stefan Vasic, and Marta Sabou. Llm-driven ontology evaluation: Verifying ontology restrictions with chatgpt. In *Data Quality meets Machine Learning and Knowledge Graphs*, 2024.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[50] Jim Webber. A programmatic introduction to neo4j. In *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity*, pages 217–218, 2012.

[51] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023.

[52] Bohui Zhang, Valentina Anita Carriero, Katrin Schreiberhuber, Stefani Tsaneva, Lucia Sanchez Gonzalez, Jongmo Kim, and Jacopo de Berardinis. Ontochat: a framework for conversational ontology engineering using language models, 2024.

Appendices

A Literature Review

The following tables describe the process steps in the creation of the structured literature review. Table 3 lists the search strings which were used, as well as the amount of screened and appropriate papers. Table 4 lists the before chosen articles.

Table 3: Google Scholar search queries

Date	Search Engine	Search Query	Screened	Appropriate
15.12.2024	Google Scholar	"Ontology engineering large language models"	30	11
15.12.2024	Google Scholar	"Large language models ontology population"	20	1
15.12.2024	Google Scholar	"Large language models ontology creation"	20	1
15.12.2024	Google Scholar	"Large language models ontology improvement"	20	1
15.12.2024	Google Scholar	"Large language models ontology validation evaluation"	10	1
15.12.2024	Google Scholar	"Large language models novice ontology engineers"	20	2
15.12.2024	Google Scholar	"Ontology engineering large language model supported framework"	20	1
15.12.2024	Google Scholar	"Ontology engineering novice developer"	20	0
15.12.2024	Google Scholar	"Ontology engineering competency questions large language models"	20	1
15.12.2024	Google Scholar	"Large language model limitations ontologies domain-specific reasoning"	20	2
15.12.2024	Google Scholar	"Ontology engineering impact LLM-supported tools"	20	1
		Total	**220**	**22**
Disclaimer: Duplicate results will be counted in the first search term they occur.				

Table 4: Overview of articles used for the literature review

Title	Reference
Ontologies in the era of large language models - a perspective	[31]
OntoChat: a Framework for Conversational Ontology Engineering using Language Models	[52]
NeOn-GPT: A Large Language Model-Powered Pipeline for Ontology Learning	[13]
LLMs4OL: Large Language Models for Ontology Learning	[14]
Investigating Open Source LLMs to Retrofit Competency Questions in Ontology Engineering	[2]
Ontology engineering with Large Language Models	[26]
From human experts to machines: An LLM supported approach to ontology and knowledge graph construction	[23]
Navigating Ontology Development with Large Language Models	[42]
Ontology Population using LLMs	[32]
LLM-driven Ontology Evaluation: Verifying Ontology Restrictions with ChatGPT	[48]
Ontogenia: Ontology Generation with Metacognitive Prompting in Large Language Models	[25]
Knowledge Engineering using Large Language Models	[4]
OLaLa: Ontology Matching with Large Language Models	[20]
LLMs4OM: Matching Ontologies with Large Language Models	[15]
Large Language Models and Knowledge Graphs: Opportunities and Challenges	[37]
A RAG Approach for Generating Competency Questions in Ontology Engineering	[38]
Towards Ontology-Enhanced Representation Learning for Large Language Models	[41]
Accelerating Knowledge Graph and Ontology Engineering with Large Language Models	[45]

Title	Reference
Leveraging LLM for Automated Ontology Extraction and Knowledge Graph Generation	[1]
Leveraging LLMs for Collaborative Ontology Engineering in Parkinson Disease Monitoring and Alerting	[8]
Dynamic Retrieval Augmented Generation of Ontologies using Artificial Intelligence (DRAGON-AI)	[47]
Large Language Models for the Creation and Use of Semantic Ontologies in Buildings: Requirements and Challenges	[28]

B Online questionnaire

The following section holds the online questionnaire template.

Usage of Large Language Models for Ontology Engineering

Thank you for participating in this questionnaire, which is part of my bachelor thesis exploring how beginners in ontology engineering utilize applications supported by Large Language Models (LLMs). The purpose of this study is to gather qualitative insights into how LLM-powered tools are used when learning ontology engineering tasks such as ontology creation, refinement, and evaluation.

Participants in the survey will be entered into a draw to win a 50 euro voucher. The winner can choose the type of voucher as long as it is available in Austria.

Please add a valid email adress where i am able to contact you.

!Disclaimer!

The data collected from this survey will be used solely for academic purposes, specifically for the completion of my thesis. All responses will be kept confidential and used only for research and analysis related to the study. No personal data will be shared with third parties.

Any personal information collected during the survey will be securely stored and will be deleted upon the successful submission and evaluation of the thesis. By participating in this survey, you consent to the use of your data as described above. Thank you for your valuable contribution.

* Gibt eine erforderliche Frage an

1. E-Mail-Adresse *

2. 1) Did you use any of the popular Large Language Models when learning Ontology engineering? *

Markieren Sie nur ein Oval.

- ( ) Yes *Fahren Sie mit Frage 4 fort*
- ( ) No *Fahren Sie mit Frage 3 fort*

Answer No

3. 2) If you did not use any LLMs, explain why? *

____

____

____

____

____

Fahren Sie mit Frage 21 fort

1) answer Yes

4. 3) Which widely known popular models did you use for Ontology Engineering? *

Wählen Sie alle zutreffenden Antworten aus.

- [ ] OpenAI ChatGPT
- [ ] Anthropic Claude
- [ ] Google Gemini
- [ ] Microsoft Copilot
- [ ] None
- [ ] Sonstiges: ______________________

5. 4) Have you heard about any LLMs or LLM-supported frameworks specifically built for Ontology Engineering? *

Markieren Sie nur ein Oval.

- Yes
- No

6. 5) Which LLM-supported frameworks for Ontology Engineering have you heard of?

Wählen Sie alle zutreffenden Antworten aus.

- [ ] NeOn_GPT
- [ ] OntoChat
- [ ] DeepOnto
- [ ] LLMs4OL
- [ ] None
- [ ] Sonstiges: ______________________

7. 6) What specific tasks within Ontology Engineering have you attempted to complete using LLMs? *

Wählen Sie alle zutreffenden Antworten aus.

- [ ] Requirements Elicitation: Ontology-based Application Idea Generation
- [ ] Requirements Elicitation: Competency Question Generation
- [ ] Ontology Creation: Generation of classes and relations
- [ ] Advanced Ontology Definitions: Generation of property characteristics and constraints
- [ ] Ontology Population: Instances generation
- [ ] Ontology improvement: Detection of inconsistencies and other mistakes
- [ ] Reasoning: Inferring Statements
- [ ] Querying: Generation of SPARQL queries
- [ ] Sonstiges: ______________________

8. 8) How effective was the LLM in generating an idea for an application relying on an ontology? *

Markieren Sie nur ein Oval.

- Very effective — generated creative and relevant application ideas
- Effective - suggested useful application ideas but required refinement
- Slightly effective - produced some generic or partially relevant ideas
- Not effective - failed to suggest meaningful or applicable ideas
- No experience

9. 9) How effective was the LLM in generating competency questions for your ontology project? *

Markieren Sie nur ein Oval.

- Very effective - generated relevant questions automatically
- Effective - required refinement but saved time
- Slightly effective - generated generic questions that needed major adjustments
- Not effective - questions were unrelated or unusable
- No experience

10. 10) How well did the LLM assist in defining properties and relationships in your ontology? *

Markieren Sie nur ein Oval.

- Very well - properties and relations were generated with minimal changes
- Somewhat well - basic properties were created, but complex ones needed input
- Not very well - only simple properties were generated correctly
- Poorly - properties and relationships were incorrect or missing
- No experience

11. 11) When defining ontology classes, how well did the LLM suggest accurate class hierarchies? *

Markieren Sie nur ein Oval.

- Very well - suggested correct and logically structured class hierarchies
- Somewhat well - with minor manual adjustments
- Not very well - significant revisions were needed
- Poorly - hierarchies were incorrect or illogical
- No experience

12. 12) How effective was the LLM in generating property characteristics and ontology restrictions? *

Markieren Sie nur ein Oval.

- Very effective - accurately defined properties and applied relevant restrictions
- Effective - generated useful property definitions but required manual refinement
- Slightly effective — suggested basic properties but failed to capture complex restrictions or properties
- Not effective - produced incorrect or irrelevant properties and restrictions
- No experience

13. 13) Did you use LLMs to create instances for your ontology? If so, how useful was this feature? *

Markieren Sie nur ein Oval.

- Very useful - generated correct instances automatically
- Useful - with some manual corrections
- Slightly useful - required significant adjustments
- Not useful - instances were inaccurate or irrelevant
- No experience

14. 14) How effective was the LLM at extracting terms from unstructured data to populate the ontology? *

Markieren Sie nur ein Oval.

- Very effective - correctly identified relevant terms
- Effective - terms needed some refinement
- Slightly effective - only a few terms were correctly extracted
- Not effective - irrelevant or unrelated terms were generated
- No experience

15. 15) How useful was the use of LLMs to validate the consistency or logical soundness of generated ontologies? *

Markieren Sie nur ein Oval.

- Very useful — detected logical errors and provided accurate validation suggestions
- Useful — identified some inconsistencies, but manual validation was still required
- Slightly useful - occasionally provided helpful insights but missed critical logical issues
- Not useful - failed to detect logical inconsistencies or provided incorrect validation results
- No experience

16. 16) If you read this sentence, pick 25. *

Markieren Sie nur ein Oval.

- 24
- 26
- 25
- 27

17. 17) How effective was the LLM in reasoning tasks such as the inference of additional triples? *

Markieren Sie nur ein Oval.

- Very effective - additional triples were correctly inferred with minimal errors
- Effective - some relevant triples were inferred but required corrections
- Slightly effective - only a few correct triples were inferred, with significant manual adjustments needed.
- Not effective — additional triples were not inferred or were incorrect
- No experience

18. 18) Did you find LLMs useful in generating SPARQL queries for querying your ontology? *

Markieren Sie nur ein Oval.

- Very useful - queries were generated accurately
- Useful - queries required some corrections
- Slightly useful - only basic queries were generated correctly
- Not useful - queries were often syntactically incorrect or irrelevant
- No experience

19. 19) What general challenges did you face when generating ontology concepts using LLMs? *

Wählen Sie alle zutreffenden Antworten aus.

- [ ] Difficulty managing context-specific terms
- [ ] Lack of precision in domain-specific knowledge
- [ ] Problems with logical consistency in generated concepts
- [ ] All of the above
- [ ] No experience

20. 20) How did you handle incorrect or incomplete outputs generated by LLMs? *

Wählen Sie alle zutreffenden Antworten aus.

- [ ] Corrected the outputs manually
- [ ] Used iterative prompting or query refinement
- [ ] Combined LLM suggestions with human input
- [ ] Avoided using LLMs for critical ontology tasks
- [ ] No experience

Interview canditates

I think it is also interesting to have a canditate interviewed who didnt use any LLMs for ontology engineering to also get some insight why they didnt use it.

21. 21) Would you be available to take part in a further semi-structured interview round? The chance on winning a 50 € voucher of your choice triples (You will be entered in the drawing pot three times). *

Markieren Sie nur ein Oval.

- ( ) Yes
- ( ) No

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt.

Google Formulare

C Semi structured interviews

The following section holds the one-pager which was sent to participants and the interview guideline which was used for the interviews.

Large Language Model usage when learning Ontology Engineering

The rapid evolution of advanced intelligent applications, particularly with the development of large language models (LLMs) such as OpenAI's ChatGPT and their versatile and modular capabilities have opened new paths when performing knowledge intensive tasks [3]. Recent advancements suggest that LLMs could play a key role in ontology engineering (OE) by providing computational support in areas such as conceptual modeling and ontology learning (OL) [1][2]. OE is a complex and systematic process to create, review and reuse structured representations of knowledge known as ontologies [1]. This process traditionally requires substantial manual effort, mainly in requirement collection and validation, leading to resource-intensive workflows [1]. Researchers aim to streamline these tasks by leveraging LLMs for generating competency questions (CQ), creating ontology drafts and validating ontology requirements [1][2].

This thesis aims to explore the potential role of LLMs in OE. The study focuses on providing an overview of existing models that can support tasks within OE, while also investigating how they are utilized in practice by novice engineers. To achieve our goals, we will conduct a literature review focused on current approaches and methodologies for using LLMs in OE, outlining their application and potential to enhance various processes within OE. Furthermore, an exploration on how novice ontology engineers make use of these tools, gaining deeper insights into their LLM-supported approaches to OE.

The research questions of the thesis are:

RQ1: How do novice ontology engineers make use of large language model supported tools when performing ontology engineering tasks?

RQ2: How do these tools assist novice engineers to understand the general domain of ontology engineering?

RQ3: How do these tools influence the overall approach to ontology engineering and in which parts of ontology engineering large language models are primarily used?

Process Understanding: Semi Structured Interviews

To gain deeper insights into novice engineers' use of large language models (LLMs) for ontology engineering (OE) tasks, semi-structured interviews (SSI) will be conducted. The semi-structured interviews will explore participants' experiences with LLM-supported tools, focusing on specific tasks identified in the survey, such as those mentioned in question six. Participants will be asked to elaborate on their workflows, challenges, and the tools they utilized or avoided. This phase aims to collect detailed insights into how LLMs are integrated into OE processes, their strengths and limitations, and how they support learning.

The goal of the semi-structured interviews, in combination with the literature review and survey, is to comprehensively address the three research questions. By integrating insights from the interviews with findings from the literature review and survey, we draw conclusions on how novice ontology engineers are supported by LLMs when learning ontology engineering.

References:

[1] Bohui Zhang, Valentina Anita Carriero, Katrin Schreiberhuber, Stefani Tsaneva, Lucia Sanchez Gonzalez, Jongmo Kim, and Jacopo de Berardinis. Ontochat: a framework for conversational ontology engineering using language models, 2024.

[2] Nadeen Fathallah, Arunav Das, Giorgis De Giorgis, Andrea Poltronieri, Peter Haase, and Liubov Kovriguina. Neon-gpt: A large language model-powered pipeline for ontology learning. In Special Track on Large Language Models for Knowledge Engineering, Extended Semantic Web Conference, 2024. (ESWC 2024), 2024.

[3] Fabian Neuhaus. Ontologies in the era of large language models – a perspective. Applied Ontology, 18(4):399–407, 2023.

Interview Guide: Semi-Structured Interviews

Priority of Questions:

1. **Essential**
2. **Optional if time**
3. **Not essential only if after completion of above ranked questions there is still time**

#	Priority	Questions
		Introductory Section
1	1	Greet the participant and thank them for their time. Ask them if they are ok with recording the interview for transcription.
2	1	Introduction to the topic. Present our research question and overall topic
		Background and Experience
3	1	Can you describe your experience with ontology-related tasks?
4	1	Have you used Large Language Models (LLMs) in any form during your ontology work? If yes, which ones?
5	2	When you first encountered LLM-based ontology tools, what were your expectations regarding their capabilities?
		LLM Tool Usage and Practices
6	1	Please walk me through a specific ontology engineering task where you used LLM-supported tools? Describe the workflow.
7	1	How did the tool assist you in different stages of ontology creation? (specialized on what they choose in question 6 of the questionnaire)
8	2	Did the tool provide any recommendations or insights that you found particularly surprising or helpful?
9	1	Can you describe any limitations you encountered when using these tools?
10	2	How did you handle inaccuracies from the LLM-generated outputs?
		Process Understanding
11	1	How did the tool influence your understanding of core ontology concepts (e.g. class hierarchies, properties, constraints)?
12	1	Did the tool enhance your efficiency in completing specific tasks (e.g., defining instances, validating consistency)? Can you elaborate?
13	2	Were there moments when you relied on additional sources or expert validation(lecturer) to confirm LLM-generated outputs? If yes, why?
		Reflective and Wrap-Up Questions
14	1	Looking forward, how do you think LLM-based tools could be improved to better support ontology engineers?
15	2	Why do you think no one from the course uses specialized frameworks for ontology engineering rather than publicly known LLMs?
16	1	Conclude by thanking the participants and explaining the next steps, such as anonymization and follow-up contact details. (if needed)