



RESEARCH ARTICLE

Automating agentic collaborative ontology engineering with role-playing simulation of LLM-powered agents and RAG technology

Andreas Soularidis¹, Dimitrios Doumanas¹, Konstantinos Kotis¹, and George A. Vouros²

¹Department of Cultural Technology and Communications, Intelligent Systems Lab, University of the Aegean, University Hill, 81100 Lesvos, Greece

²Department of Digital Systems, AI Lab, Gr. Lampraki 126, University of Piraeus, Piraeus, Greece

Corresponding author: Andreas Soularidis; Email: soularidis@aegean.gr

Received: 5 February 2025; **Revised:** 20 November 2025; **Accepted:** 21 November 2025

Abstract

Motivated by the astonishing capabilities of large language models (LLMs) in text-generation, reasoning, and simulation of complex human behaviors, in this paper, we propose a novel multi-component LLM-based framework, namely LLM4ACOE, that fully automates the collaborative ontology engineering (COE) process using role-playing simulation of LLM agents and retrieval augmented generation (RAG) technology. The proposed solution enhances the LLM-powered role-playing simulation with RAG ‘feeding’ the LLM with three different types of external knowledge. This knowledge corresponds to the knowledge required by each of the COE roles (agents), using a component-based framework, as follows: (a) domain-specific data-centric documents, (b) OWL documentation, and (c) ReAct guidelines. The aforementioned components are evaluated in combination, with the aim of investigating their impact on the quality of generated ontologies. The aim of this work is twofold, (a) to identify the capacity of LLM-based agents to generate acceptable (by human-experts) ontologies through agentic collaborative ontology engineering (ACOE) role-playing simulation, at specific levels of acceptance (accuracy, validity, and expressiveness of ontologies) without human intervention and (b) to investigate whether and/or to what extent the selected RAG components affect the quality of the generated ontologies. The evaluation of this novel approach is performed using ChatGPT-o in the domain of search and rescue (SAR) missions. To assess the generated ontologies, quantitative and qualitative measures are employed, focusing on coverage, expressiveness, structure, and human involvement.

1. Introduction

The engineering of ontologies (OE) is fundamental for delivering structured knowledge in formal and explicit ways, enabling semantic interoperability, integration, and linking of data across different systems and applications. Collaborative ontology engineering (COE) involves multiple stakeholders, such as Domain Experts, Knowledge Engineers, and so called, Knowledge Workers. A collaborative and iterative OE approach ensures that engineered ontology captures diverse perspectives and insights, resulting in a more comprehensive and accurate representation of the domain. The importance of COE lies in its ability to integrate collective expertise iteratively and continuously, which is crucial for addressing the complexity and intricacy of real-world domains.

Our motivation to conduct research on advancing COE through artificial intelligence (AI)-assisted agent-based automation stems from its inherent challenges and demands, as well as from the strengths

Cite this article: A. Soularidis, D. Doumanas, K. Kotis and G. Vouros. Automating agentic collaborative ontology engineering with role-playing simulation of LLM-powered agents and RAG technology. *The Knowledge Engineering Review* 40(e10): 1–53. <https://doi.org/10.1017/S026988892510009X>

© The Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike licence (<https://creativecommons.org/licenses/by-nc-sa/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the same Creative Commons licence is used to distribute the re-used or adapted article and the original article is properly cited. The written permission of Cambridge University Press or the rights holder(s) must be obtained prior to any commercial use.



of LLMs to support knowledge-intensive tasks. COE is labor-intensive, time-consuming, and prone to incomplete representation of knowledge (Glimm *et al.*, 2014). Furthermore, it necessitates extensive expertise and meticulous attention to detailed definitions of ontological axioms. Recent related works on the automation of COE (Doumanas *et al.*, 2025) demonstrated potential solutions by enhancing the ontology development process, reducing the burden on human experts, and increasing efficiency. However, while automation can significantly enhance productivity, it is not without limitations. Automated approaches may lack the nuanced understanding of the domain and the contextual awareness that human experts provide. Therefore, human involvement remains crucial, at least for refining and validating the generated ontologies, ensuring their accuracy, validity, and expressiveness.

Collaboration and role-playing simulation play a pivotal role in this context. Role-playing in agentic collaborative ontology engineering (ACOE) involves simulating various stakeholder roles, allowing participants to adopt different perspectives and better understand the needs and constraints of each role. This approach improves communication and enhances problem-solving capabilities within the team. By combining automation and collaborative role-playing, the strengths of both can be leveraged.

The advent of LLMs such as OpenAI's GPT4, Anthropic's Claude, etc., have opened new avenues in a plethora of fields including OE (Doumanas *et al.*, 2024). LLMs are models trained on vast amounts of data, showing a remarkable performance on generating human-like text, making them powerful tools for automating various aspects of OE, such as translating natural language (NL) into SWRL (Soularidis *et al.*, 2024), text-to-SPARQL (Avila *et al.*, 2024), ontology learning (Lo *et al.*, 2024), etc. The application of LLMs in OE has the potential to markedly reduce the time and effort required for delivering domain ontologies. Furthermore, the potential domain knowledge of LLMs, as acquired through exposure to related data during training, could further enrich and improve the quality of the generated ontologies. In addition, LLMs have demonstrated remarkable proficiency in role-playing, empowering various simulations (Park *et al.*, 2023). Consequently, the integration of role-playing and LLMs has the potential to further enhance COE, towards the new concept of ACOE.

On the other hand, LLMs have been observed to frequently exhibit irrelevant (Reddy *et al.*, 2024), erroneous and outdated responses (Liu *et al.*, 2023), factors that diminish their credibility and confidence. Retrieval Augmented Generation (RAG) technology has emerged in the field with the objective of reducing the aforementioned limitations. RAG can enhance the fine-tuning of responses generated by LLMs as it facilitates the augmentation of prompts with additional (external) information (Salemi *et al.*, 2024). This functionality enables the feed of LLMs with domain-focused specific data, achieving more up-to-date and accurate responses (from generic to narrow knowledge), together with other task/problem specific information. Subsequently, it can be conjectured that RAG is a promising technological solution with the potential to play a pivotal role in the OE process.

In this context, the paper presents a novel LLM-based framework, namely LLM4ACOE, that automates the COE process via role-playing simulation and RAG technology, extending our previous work on a fully automated LLM-based COE methodology, namely the Sim-HCOME methodology (Doumanas *et al.*, 2025). Particularly, the LLM is prompted to collaboratively engineer ontologies by simulating three roles (one per agent: Knowledge Engineer, Domain Expert, Knowledge Worker) defined in the HCOME methodology (Kotis & Vouros, 2006; Paparidis & Kotis, 2021), following an iterative discussion. The LLM user, a designated Prompt Engineer (human agent), is tasked with the crafting of the prompt, ensuring the incorporation of all essential ontology-specific information, including the aim, scope, requirements, and Competency Questions (CQs). In addition to this, the Prompt Engineer fulfills the role of human expert, a further role beyond the simulated ones, responsible for the evaluation of the engineered ontologies, without the capacity to intervene in the ACOE process. To support agents in playing their simulated roles, in this paper, we leverage RAG technology to feed the LLM with additional information, in addition to the information provided in the prompt that is, the aim, scope, requirements and CQs of the ontology to be created. Particularly, the proposed approach utilizes RAG technology to feed the LLM with (a) domain-specific data-centric documents and (b) the Web Ontology Language (OWL) documentation, and (c) ReAct guidelines¹ (Yao *et al.*, 2023). The evaluation of the proposed

¹The ReAct framework is not incorporated into the proposed ACOE approach, instead, the ReAct approached is followed.

approach is performed using ChatGPT4-o, after experimentation and evaluation of other state-of-the-art LLMs such as Gemini Pro and Claude Sonnet.

Summarizing, the aim of this work is twofold, (a) to identify the capacity of LLM-based agents to generate acceptable (by human-experts) ontologies, at specific levels of acceptance (accurate, valid, and expressive ontologies) through ACOE role-playing simulation, without any human interventions and (b) to investigate whether and/or to what extent the selected RAG components supporting ACOE, affect the quality of the generated ontologies. The contribution of this paper is summarized as follows:

1. It presents a novel approach to fully automated LLM-based ACOE using RAG and role-playing simulations.
2. It reports research findings related to the impact of knowledge (i.e. beyond that of LLMs) injected into the prompt via RAG components on the quality of the generated ontologies.
3. It applies and evaluates the proposed approach to the SAR domain, demonstrating its practical utility and effectiveness in a real-world scenario.

The structure of the paper is as follows: Section 2 presents background knowledge regarding the role-playing simulation, Sim-HCOME methodology, RAG technology, and ReAct framework, while Section 3 presents related work. Section 4 presents the proposed framework and implementation details, while Section 5 discusses the experimental setup. Section 6 reports the results of the experiments conducted, and Section 7 critically discusses these results. Finally, Section 8 concludes the paper and reports future plans.

2. Background

2.1. Role-playing simulation

LLMs have emerged in the field of AI demonstrating remarkable capabilities in text understanding, generation, and reasoning. These capabilities of LLM motivated people to instruct LLMs to take on roles they desire, such as movie stars, game characters, or even their own relatives (Chen *et al.*, 2024). This behavior of LLM to simulate specific characters/personas is known as role-playing. Their accomplishments extend beyond the simulation of unique personas, effectively simulating a broader range of human behaviors. Particularly, they have demonstrated more sophisticated capabilities toward anthropomorphic cognition, including humanity emulation (Chang *et al.*, 2023; Shanahan *et al.*, 2023), and social intelligence (Kosinski, 2024), producing a highly convincing sense of human likeness. These role-playing simulated agents allow us to proceed to the simulation of many real-world settings, in order to understand, analyze trends in behavior or to fully automate several engineering tasks, including OE related ones, as we aim to present in this work.

2.2. Sim-HCOME

Sim-HCOME (Doumanas *et al.*, 2025) extends the HCOME OE methodology (Kotis & Vouros, 2006; Paparidis & Kotis, 2021) by integrating the power of LLMs and role-playing simulation, where LLM-simulated agents work entirely autonomously, fully automating the OE process. In this approach, the LLM is assigned the lead role in OE tasks, simulating the three roles of HCOME that is, Knowledge Engineer, Domain Expert, and Knowledge Worker, in an iterative discussion towards the development of domain specific ontologies (Figure 1). The Prompt Engineer ‘feeds’ the model, in a single prompt, with all the necessary information needed for role-playing simulation, describing the given roles and the responsibilities that each one of them will have. The basic input information regarding the OE remains the same that is, aim and scope of the ontology, its requirements, and its CQs. The LLM, taking into account the given data, performs the simulation via an iterative discussion between the three assigned roles, trying to capture the domain knowledge and generate a coherent ontology. It is important to note that the LLM decides autonomously when the simulated COE is completed, and no further exploration

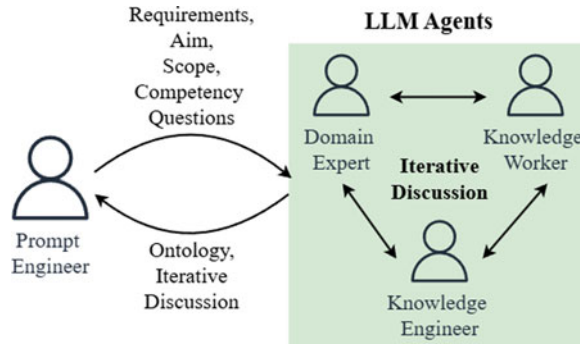


Figure 1. *The Sim-HCOME COE methodology*

or discussion between the three roles is needed, since the Prompt Engineer (human) is not allowed to interfere at any stage apart from evaluating the final outcome that is, the generated ontology. At this point, it is worth noting that we have previously explored a similar version of this approach, where human agents are also involved in ACOE. This approach, namely SimX-HCOME, is evaluated and presented in another work of ours (Doumanas *et al.*, 2025). The focus of this paper is limited to fully automated approaches powered by LLMs, role-playing simulation and RAG technology.

2.3. RAG

Despite their incredible performance in human-like text generation and their integration in various domains such as software engineering (Huang *et al.*, 2023), notaries and law (Litaina *et al.*, 2024), etc., LLMs suffer from erroneous, inconsistent, or even outdated responses (Perkovic *et al.*, 2024). RAG technology has been developed to enhance the accuracy and reliability of generative AI (GenAI) models with additional data fetched from external resources (e.g., files in various formats, data from the Web, etc.). By pulling in relevant data from external resources, RAG allows LLMs to generate more precise, accurate, and contextually appropriate responses.

In its simplest version, the RAG process can be broken down into two principal stages. In the first stage, the external data are loaded and split into smaller parts, known as chunks. These chunks are then transformed (embedded) into vectors and stored in vector stores. The second stage is the retrieval process, which aims to retrieve the most relevant data from the vector store based on the user input. To achieve this, the user input (prompt) is also transformed into a vector. A special component is used, namely the *Retriever*, aiming to find and retrieve the most similar context based on the given input (query), by measuring the distance between the user query and the data from the vector store. This is done following related algorithms and techniques, such as the cosine similarity, etc. Finally, the retrieved data, which are called documents, are pushed to the LLM (along with the prompt) to generate the response. The whole RAG process is described in Figure 2.

2.4. ReAct

Motivated by the inherent ability of humans to combine task-oriented actions with verbal reasoning, Yao *et al.* (2023) proposed the ReAct framework in language models. The proposed approach combines both reasoning traces and task-specific actions following an interleaved manner to accomplish a given goal. The synergy between the two main procedures of ‘thinking’ and ‘acting’ promote the handling of exceptions or adjustment of the initial plan according to the environment situation. Conversely to conventional ‘chain-of-thought’ approach that works as a static black box, ReAct goes beyond by prompting the LLM to generate both verbal reasoning traces and task-specific actions iteratively, allowing the LLM to perform dynamic reasoning based on the current situation and the external environment.

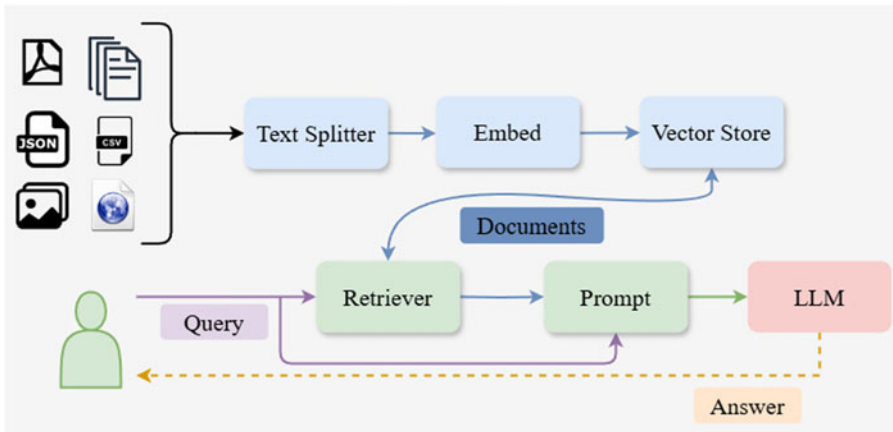


Figure 2. Basic RAG architecture

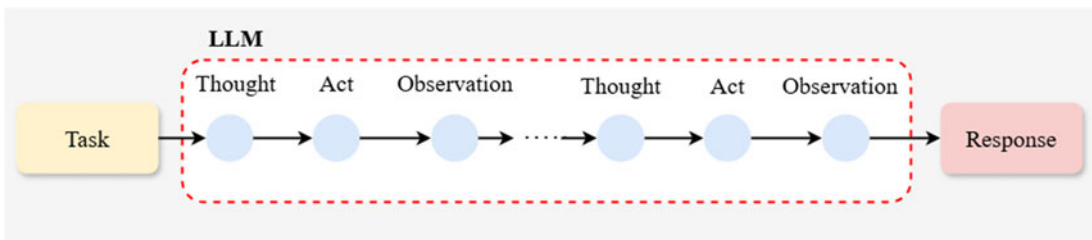


Figure 3. The ReAct approach

Their approach, given a goal, tries to decompose it into smaller tasks, following thought–act–observation iterations until the goal is accomplished, as depicted in Figure 3. To do that, the LLM is fed with manually created ReAct-format trajectories (thought-action-observation steps) that constitute randomly selected cases from a training set. The experimental results, performed on multi-hop question-answering, fact checking, and iterative decision-making tasks, demonstrate the supremacy of the proposed approach compared to traditional methods. It must be noted that in the research work presented in this paper, the ReAct framework is not incorporated as it has been originally proposed. Instead, we tailor the ReAct approach to OE tasks such as adding a new class to the ontology, reusing an existing ontology, etc. Thus, instead of referring to the ReAct framework, we refer to ReAct guidelines tailored to support OE tasks.

3. Related work

3.1. LLMs and OE

Zhang *et al.* (2024) propose an LLM-based framework, namely OntoChat, to foster collaborative ontology engineering by implementing a conversational workflow among LLMs and stakeholders (i.e., ontology engineers, domain experts). The proposed framework facilitates challenging and time-consuming manual tasks concerning the OE as they rated by ontology engineers in a survey conducted. The OntoChat leverages LLMs to facilitate requirement elicitation by creating user stories and generating CQs, analysis by providing CQs verification, reduction, and clustering, and finally testing by verbalizing the generated ontologies. The proposed framework is evaluated by replicating the engineering of the Music Meta ontology. The experimental results demonstrate, despite some misses and limitations, a positive response from the stakeholders, indicating potential for accelerating conventional ontology engineering tasks.

Fathallah *et al.* (2024) focus on synergy of LLMs with OE methodologies, integrating the NeOn OE methodology, using prompt engineering techniques, and proposing the NeOn-GPT, an LLM-based framework for ontology learning. The proposed framework leverages the power of LLMs to automatically translate natural language domain descriptions into turtle syntax ontologies. The authors capitalize prompt engineering techniques such as CoT, few-shot prompting, and role-playing to achieve more relevant responses. The proposed framework uses external online tools such as HermitT reasoner Application Programming Interface (API) (Glimm *et al.*, 2014) and OOPS API (Poveda-Villalón *et al.*, 2014) to perform syntax validation, consistency check and pitfall resolution, while it is evaluated using the Stanford wine ontology as the gold standard. The experimental results illustrate that the combination of prompt engineering techniques with OE methodologies can facilitate the generation of more consistent ontologies by the LLMs.

Mateiu and Groza (2023) present a method for enhancing ontologies by translating natural language sentences into Description Logic using a fine-tuned GPT-3 model. This approach aims to address the technical challenges and costs of ontology development, which have hindered the success of the Semantic Web. Implemented as a plugin for the Protégé editor, the tool automates the creation and enrichment of ontologies by converting NL into OWL Functional Syntax. The authors trained the model on a dataset of 150 prompt-axiom pairs, demonstrating its ability to handle various ontology-related tasks. This automation reduces the need for constant expert input, thus saving development time and improving decision-making in ontology engineering.

Doumanas *et al.* (2024) focus on a collaborative approach between humans and LLMs, proposing X-HCOME. Motivated by the capabilities of LLMs to understand enormous amount of data and generate human-like text, the proposed methodology adjusts and extends the HCOME OEM (Kotis & Vouros, 2006; Paparidis & Kotis, 2021) by integrating LLMs into the OE lifecycle in the context of SAR missions. The proposed methodology is centered on the synergy of human expertise and the capabilities of LLMs. This hybrid approach facilitates the development of ontologies addressing some of the common limitations that are inherent into conventional OEM, facilitating the rapid development of ontologies.

A recent work of Doumanas *et al.* (2025) investigates the potential of LLMs to facilitate, speed up or even automate the OE process by experimenting with distinct levels of LLM's involvement into the OE process. The proposed approach is grounded in HCOME OEM and explores the impact on the quality of the generated ontologies as human involvement is reduced. The investigation encompasses the entire range of synergy between humans and LLMs, spanning from manual human-generated ontologies to automated LLM-generated ontologies. The experiments conducted on two distinct domains, SAR missions and Parkinson's disease, have yielded noteworthy outcomes. These experiments have demonstrated the impact on LLMs in the OE pipeline, highlighting its potential to enhance both the speed and the accuracy of the process. Moreover, these experiments have underscored the necessity of human expertise, particularly in critical interventions and adjustments, thereby emphasizing the importance of human involvement in the context of OEMs with LLMs.

3.2. LLMs and RAG

Despite their remarkable performance in analyzing enormous amount of data and generating human-like text, LLMs are frequently suffer from irrelevant or outdated responses. RAG has emerged as a solution of the above limitations of LLMs, while also providing them with domain knowledge from external sources.

In this context, Pan *et al.* (2024) present a RAG-based approach to automate the generation of CQs for OE. This study combines the remarkable capabilities of LLM in text generation, enhanced with the domain knowledge available via RAG, with the aim of automating the time-consuming and labor-intense generation of CQs. The study evaluates both the quality and quantity of external resources utilized in the generation of CQs. The experimental results demonstrate the superiority of RAG in cases where more domain knowledge is required, compared to zero-shot prompting techniques. The research also highlights that not only the quantity but also the quality of external data improves the efficacy of RAG-based solutions.

Soularidis *et al.* (2024) proposes an ontology-based framework combining the power of LLMs with RAG to automatically translate rules from NL into SWRL following a template driven approach. In this study, the authors utilize RAG to feed the LLM with ontology concepts to further assist the LLM in the translation process. The experiments conducted in the domain of SAR missions demonstrate that enhancing the LLM with external knowledge via RAG and implementing prompt engineering techniques results in more accurate and well-formed SWRL rules.

Kommineni *et al.* (2024) investigate the (semi-) automatic construction of Knowledge Graphs (KGs) leveraging open-source LLMs and RAG. In this direction, the proposed approach includes a pipeline from CQs generation and Ontology creation to KG construction and evaluation. The study focuses on the construction of KGs with little or no human intervention. In this context, the RAG approach is used to facilitate the KG construction by retrieving the answers to the defined CQs from a set of scholarly publications. The experimental results obtained highlight the potential for ontology and KG generation requiring less effort and less expertise in semantic web technologies.

To address the challenge of hallucination in AI-driven question-answering (QA) systems, Zhao *et al.* (2024) propose CyberRAG. The proposed approach integrates LLMs with an ontology-aware RAG mechanism to enhance the safety and reliability of QA systems in education. This pioneering approach focuses on the quality of the generated results that are critical in such educational systems, ensuring the validity of LLMs responses even in cases where the questions fall outside the scope of the RAG's knowledge base. To prevent the LLM from responding solely on the base of its own knowledge, the proposed solution augments the LLM knowledge with domain-specific data and validates the responses by leveraging ontologies and KGs. This fosters a more interactive and secure learning environment.

DeBellis *et al.* (2024) motivated by the exposed hallucinations, bias, black-box reasoning and lack of in-depth domain knowledge that is inherent to LLMs, propose a KG RAG-based architecture to tackle the aforementioned challenges towards the development of a semantic search toll for dental products and materials. The proposed solution utilizes an integrated knowledge base for storing both the vectors and documents as well as additional contextual domain knowledge provided by the ontology. Some of the advantages of the proposed approach compared to the traditional RAG based solutions include knowledge reuse, flexibility, and improved context. The authors envision that this approach will pave the way for the integration of semantic and machine learning technology in a plethora of domains including healthcare, legal, and security.

3.3. LLMs and role-playing simulations

Hu *et al.* (2023) explore the use of LLMs as user simulators to improve task-oriented dialogue systems. The authors propose a novel approach called User-Guided Response Optimization (UGRO), which optimizes fine-tuned Task-Oriented Dialogue (TOD) models by incorporating satisfaction feedback from LLM-powered user simulators. By leveraging the capabilities of LLMs to predict satisfaction scores, the UGRO approach aims to enhance the performance of dialogue systems without the need for manual annotations. The study validates the effectiveness of UGRO through empirical experiments on TOD benchmarks, demonstrating improvements in generated responses, user satisfaction and semantic quality. The research highlights the potential of integrating LLMs as user simulators to enhance the dialogue experience and suggests future exploration of different forms of interaction between LLMs and TOD systems.

Updyke *et al.* (2023) explore the integration of LLMs in simulation frameworks to enhance the realism of human behavior modeling. By incorporating factors such as knowledge acquisition, motivations based on the Reiss Motivation Profile, and interpersonal relationships among simulated agents, the research aims to create more authentic interactions. The study highlights the potential of LLMs in guiding virtual agents to perform coherent and realistic tasks, with implications for improving simulations in fields like cybersecurity and social media analysis. Future research directions include optimizing agents understanding and response to LLM directives to further enhance the fidelity of simulated human activities.

Gui and Toubia (2023) examine the complexities of using LLMs like GPT to simulate human behavior for experimental research. It highlights the challenges of ensuring that simulated patterns can be interpreted as casual, emphasizing issues with confounding factors when varying treatments in LLM prompts. The paper reviews traditional experimental methods and discusses how LLMs can mimic these, proposing two approaches to address confounders: adding detailed controls and specifying experimental designs. A theoretical framework is presented to understand and potentially overcome these challenges, concluding that while LLMs have significant potential, ensuring valid cause-and-effect relationships in simulations requires further research and methodological advancements.

Argyle *et al.* (2022) explore, using GPT-3, the possibility of using LLMs as a proxy for human sub-populations in social science research. The authors introduce the concept of ‘algorithmic fidelity’, showing that with proper conditioning on socio-demographic backstories, GPT-3 can emulate nuanced human response distributions accurately. This allows for the creation of ‘silicon samples’ that reflex complex human attitudes and socio-cultural contexts. The study demonstrates the potential of GPT-3 in tasks such as partisan text generation, vote prediction, and survey responses, highlighting its promise as a powerful tool for advancing social science research.

Filippas *et al.* (2024) explores the use of LLMs like GPT-3 to simulate human economic behavior for research purposes. The authors introduce the concept of ‘homo silicus’, a computational analogue to ‘homo economicus’, which can be used to conduct AI-based economic experiments. The paper demonstrates that LLMs can replicate classic behavioral economics experiments, providing similar insights to those obtained from human subjects. These simulations can serve as cost-effective and rapid preliminary studies, guiding real-world empirical research by exploring a wide range of scenarios and parameters. While emphasizing the utility of LLMs in economic research, the author acknowledges that findings from AI experiments require empirical validation with actual human behavior.

3.4. Summary and conclusion

Current LLM-enhanced OE/COE approaches are facing limitations, beyond those that are inherent to LLMs. These limitations have a direct impact on both the quality of the generated ontologies and on the OE process in general. These include the restricted or even, in certain instances, outdated domain knowledge in specialized domains, which has a deleterious effect on the ontology coverage and the knowledge that represents (incomplete knowledge). This limitation also leads to an increased reliance on human expertise, thereby undermining the entire OE process (Zhang *et al.*, 2024). Additionally, it has been demonstrated that LLMs are deficient in their ability to transform their inherent knowledge into coherent and well-formed ontologies. This deficiency leads to ontologies with poor ontological axioms (e.g., encoded in OWL) and a reduced level of expressivity in comparison to those generated by humans (Fathallah *et al.*, 2024). Finally, existing research often focuses on isolated aspects, such as CQ generation, text-to-SWRL, and text-to-SPARQL translation, rather than integrated/complete solutions.

The aim of this research is to address the aforementioned limitations by proposing a multi-agent LLM-based framework for ACOE, namely LLM4ACOE, which automates the COE process through role-playing simulation of LLM agents and RAG technology. The proposed comprehensive, multi-agent LLM-based framework is capable of expanding the knowledge of LLMs with regard to both domain knowledge and OE aspects, by leveraging RAG technology. This research evaluates the impact of additional information, injected to the LLMs via RAG components, on the quality of the generated ontologies engineered through role-playing simulation, while, secondly, assess the effectiveness of state-of-the-art RAG-enhanced LLMs in automating the entire ACOE process, focusing on accuracy, validity, coverage, and expressiveness.

In summary, the proposed agentic COE approach addresses the inherent limitations of LLMs concerning the domain knowledge and COE aspects by enhancing agents participating in role-playing simulation with additional information, via RAG, such as domain-specific data (of real-world SAR missions), OWL documentation (examples of OWL axioms), and ReAct guidelines (thought-action-observation trajectories) tailored to OE tasks. The goal of this additional information is to foster the

agents' capability to generate well-structured, coherent, and expressive domain-specific ontologies, which also demonstrate a high level of domain coverage.

By applying the proposed LLM4ACOE framework to the SAR domain, its practical utility in a real-world scenario is evaluated and demonstrated.

4. The LLM4ACOE framework

4.1. Introduction

As stated already, the aim of the LLM4ACOE framework is to automate the COE process with LLM-powered RAG-enhanced agent-based role-playing simulation. Towards this direction, RAG technology is utilized, constituting the core of the framework, allowing engineers to add external knowledge such as domain-specific data, while also feeding the LLM with ontology engineering guidelines and ontological axioms examples derived from the OWL documentation, together with ReAct guidelines.

LLM4ACOE is implemented using LangChain², a widely used, open-source Python library that facilitates the development of applications powered by LLMs. LangChain provides the necessary functionality and a high degree of customizability offering plenty of models (e.g., GPT-4o, Claude, Gemini, etc.), vector stores, retrievers and retrieval algorithms, prompt templates, etc. Furthermore, LangChain incorporates external platforms such as LangGraph³ and LangSmith⁴. Specifically, the LangSmith platform offers functionality for the monitoring of LangChain-based applications. A key function of LangSmith is the provision of valuable data regarding RAG. Specifically, it offers comprehensive monitoring of the application, providing detailed information on the documents retrieved based on user input, the number of tokens consumed during a prompt-response transaction, execution times, the total cost of each transaction, and other relevant data. The LLM4ACOE framework, the prompts and the external data used to test and evaluate it, and the experimental results obtained from our evaluation strategy, are available in a GitHub repo⁵.

4.2. Framework architecture

As depicted in Figure 4, the framework architecture is composed of different types of modules, each one meticulously designed to execute a discrete functionality towards automating the COE process through role-playing simulation. Specifically, the framework is injected with domain-specific data, OWL documentation and ReAct guidelines, via RAG, and a prompt, and it generates an iterative discussion via role-playing simulation, resulting eventually to the generation of an ontology in Turtle (ontology.ttl) format, along with the generated arguments (discussion between the three different roles assigned to the LLM) in a text (discussion.txt) format.

4.2.1. RAG components

As already pointed out, the LLM4ACOE framework consists of three RAG components, supporting the LLM with three different types of information that is, domain-specific data, OWL documentation, and ReAct guidelines on OE tasks. Each RAG component is designed to support/enhance particular COE roles played by LLM agents: Knowledge Engineer, Domain Expert, and Knowledge Worker, with the objective of developing coherent, well-structured, and expressive ontologies capturing the domain knowledge adequately according to human evaluators (experts).

To ensure that the LLM is fed with data collected from all three RAG components, three Vector Stores and three Retrievers have been implemented, one for each RAG component. Each one of those components is engineered to retrieve the most similar data from the related vector stores based on Prompt

²<https://python.langchain.com/docs/introduction/>.

³<https://www.langchain.com/langgraph>.

⁴<https://www.langchain.com/langsmith>.

⁵<https://github.com/AndreasSoularidis/LLM-based-OE-Framework-LC3>.

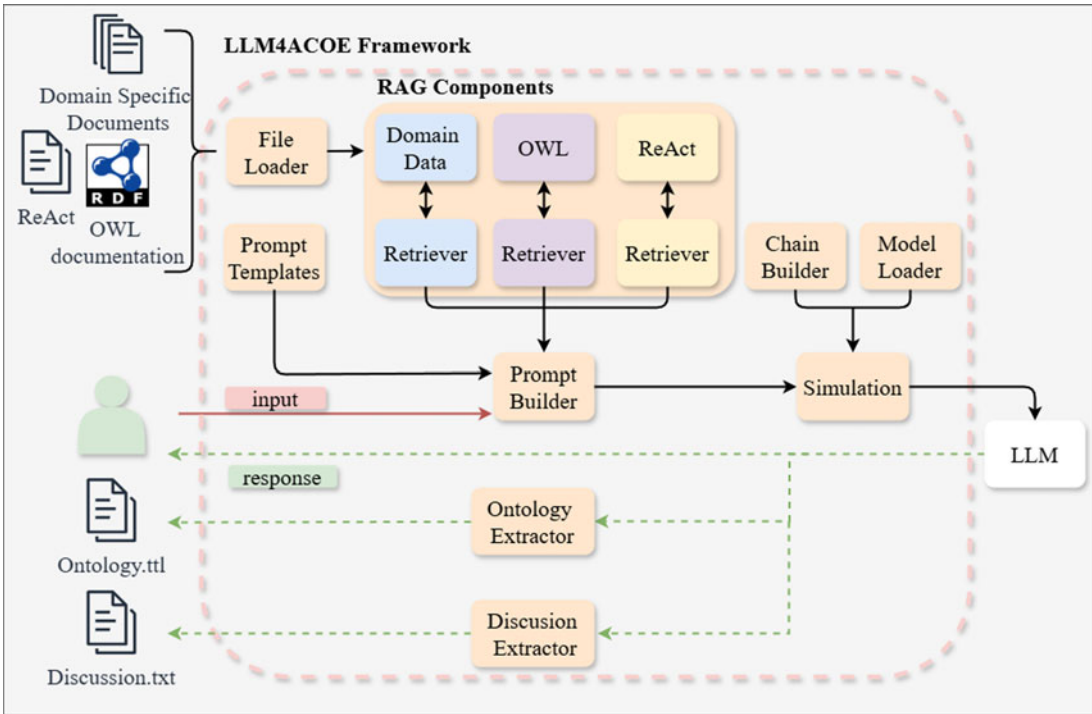


Figure 4. *Proposed framework architecture*

Engineer input. Currently, LLM4ACOE supports various data type formats such as text, pdf and csv files, while it is also capable of getting data from webpages implementing Web scraping techniques. The three RAG components are described in the following paragraphs:

- The first RAG component (Figure 4: Domain Specific Documents) feeds the LLM with domain specific data (e.g., SAR documents collected from real-world scenario). The aim of this component is twofold: firstly, to enhance the knowledge of Domain Expert and Knowledge Worker simulated agents with domain-specific information, towards enriching the generated ontologies with ontology elements for example, classes, object properties, etc., and secondly, to enhance LLMs to the ‘discovering’ of domain-related elements based on the data that it has been exposed to during its training phase.
- The second RAG component (Figure 4: OWL documentation) provides the Knowledge Engineer and Knowledge Worker simulated agents with examples and guidelines concerning OWL axioms, as these are described in the OWL reference documentation. This data is scraped from the OWL documentation webpage⁶ and aims to enhance the agent’s expertise in OE aspects to generate expressive and well-structured ontologies using OWL axioms through the role-playing simulation.
- Finally, the third RAG component (Figure 4: ReAct) provides the Knowledge Engineer simulated agent with reasoning traces and task-specific actions regarding OE, following the ReAct approach. Towards this direction a set of reasoning traces (following the thought-action-observation pattern) that describe the OE process are carefully crafted. Particularly, four documents have been engineered for the description of OE tasks: (a) the creation of a new ontology, (b) the integration of a new class into the ontology, (c) the integration of a new

⁶<https://www.w3.org/2007/OWL/draft/owl2-primer/>.

Question: How should a new ontology be created to meet the given aim and scope?

Thought: To avoid duplicating existing knowledge, I should first check if there are established ontologies relevant to this domain.

Action: Search the web for ontologies related to this domain.

Observation: I have found several related ontologies in this area.

Thought: To keep the ontology focused and practical, I should only consider the most widely used and currently maintained (live) ontologies.

Action: Review the list of existing ontologies and select only those that are widely used and currently maintained.

Observation: Now I have a refined list of widely used, live ontologies for reference.

Thought: Next, I need the IRIs of these selected ontologies to integrate them effectively.

Action: Perform a web search to locate the IRIs of the selected ontologies.

Observation: I have obtained the IRIs of the relevant ontologies, which are ready for integration.

Response: I will proceed by reusing these widely adopted, live ontologies in the new ontology, using their respective IRIs and prefixes to ensure compatibility and reusability.

Figure 5. Example screenshot of reasoning traces following the ReAct approach tailored to OE methodology

object property and (d) the integration of a new data property. An example of these traces is depicted in Figure 5. The goal of this component is to guide the LLM during the simulated role-playing iterative discussion, to ‘think’ and ‘act’ following the given reasoning traces, and to perform the necessary OE tasks towards the development of a coherent, comprehensive, and well-structured ontology. The LLM is prompted to follow the described reasoning traces behind the scenes, without showing them to the prompt engineer, as the ReAct framework does, to keep the iterative discussion as coherent and clear as possible.

The LLM4ACOE framework enables Prompt Engineers to select the preferred RAG components, thereby allowing the following combinations: (a) all components enabled, meaning that the LLM is fed with all the available types of information (i.e., domain-specific data, OWL documentation, and ReAct guidelines), (b) two components enabled: domain-specific data and ReAct guidelines, (c) one component enabled: ReAct guidelines, and (d) no component enabled: meaning that the LLM is without any external data to perform the OE process (bare LLM). The aforementioned configurations of LLM4ACOE are comparatively evaluated to study the necessity of each of the RAG components, as described in Section 5.

4.2.2. Framework modules

In addition to the RAG technology integrated in LLM4ACOE, a number of additional modules have been developed to support the COE process. According to the selected combination of RAG components, the *Prompt Builder* module is responsible for building the prompt, following a template-driven approach (as described in Section 4.3). The *Prompt Template* module is responsible for holding and providing the available prompt templates accordingly. Specifically, the Prompt Builder module receives the appropriate templates from the Prompt Templates module, the retrieved documents from the Retrievers, the input from the prompt engineer, and then it builds the prompt.

The *Chain Builder* module is responsible for creating the workflow (chain of actions/tasks) that is followed, based on the selected combination of RAG components. Furthermore, the *Model Loader* module facilitates the selection and configuration (e.g., the temperature hyperparameter) of the desired LLM by the Prompt Engineer. Finally, the *Simulation* module is used to synthesize the results of the aforementioned modules, and performs the role-playing simulation by invoking the API of the selected LLM.

Finally, the generated responses are formatted and presented to the Prompt Engineer. It is important to note that the LLM4ACOE framework employs a one-shoot prompting approach and is not equipped with memory, since a human agent is not involved in the ACOE process. Consequently, the Prompt Engineer (human agent/expert) is not able to contribute in the ACOE process, in contrast to the SimX-HCOME OEM (Doumanas *et al.*, 2025). Subsequently, the evaluation of the resulting ontologies by a human agent (expert in OE) is not part of the proposed LLM4ACOE framework, and it is performed separately using third-party tools such as Protégé. On the other hand, the LLM4ACOE implements the *Ontology Extraction* module to extract the generated ontology from the LLM's response and store it in Turtle format. Finally, the *Discussion Extractor* module is used to store the iterative discussion, including the generated ontology, in a text file.

4.3. Prompt engineering

The initial (injected) prompt, as illustrated in Appendix A, is comprised of six sections following a template-driven approach. These sections have been designed based on the best practices and our previous experience working with LLMs and OE (Doumanas *et al.*, 2025). The first section furnishes the context, encompassing the delineation of the designated roles to the LLM, their respective responsibilities and duties within the OE process, whilst offering a concise description of the HCOME OE methodology. The second section constitutes the prompt engineer input, in which the ontology requirements aim, scope, and CQs are described. The subsequent three sections pertain to the RAG components (one for each component) and are populated by the corresponding retrievers with data based on the user input. Subsequently, the prompt consists of three placeholders in the form of *{RAG-data}*, one for each section/retriever, where the retrieved documents are placed. It is noted that these three sections are optional and are populated based on the selected RAG components. Finally, the last section provides general guidelines regarding the iterative discussion and the format of the generated ontology.

4.4. Hyperparameters configuration

A crucial aspect that exerts a direct influence on the quality of the resulting ontologies pertains to the values of certain hyperparameters. The proposed framework empowers the Prompt Engineer with a high degree of flexibility in determining the values of some hyperparameters. The first of these concerns the LLM *temperature* that controls the randomness of text generated. The temperature ranges between 0 and 1, with lower values driving the LLM to select (predict) the next word with the highest probability, and higher values increase the likelihood of selecting less probable words.

Chunk size constitutes another significant hyperparameter that pertains to the maximum number of characters that a chunk should contain when splitting the external data in RAG pipeline. Increasing the value of this parameter results in larger chunks, thereby increasing the probability of retrieving various information via RAGs. Conversely, decreasing this value leads to the creation of more and smaller chunks, which may result in chunks containing insufficient or even misleading information.

Chunk overlap defines the number of characters that should overlap between two adjacent chunks. Increasing the value of chunk overlap results in chunks sharing more common data, whereas smaller values result in chunks sharing less or even no common data.

Lastly, it is important to note that the *number of retrieved documents (chunks)* via RAG can have a substantial impact on the quality of the results obtained. The aforementioned hyperparameters have a

significant impact not only on the quality of LLM responses, but also on the total number of tokens used (and consequently the total cost) during a prompt-response transaction.

Finally, the *search type* used by the retriever can also have a significant impact on the quality of the results obtained. At present, LLM4ACOE supports two types of searches. Specifically, the *similarity search* method compels the retriever to select chunks that are most similar to the input. Conversely, the *maximal marginal relevance* (MMR) method selects the most similar data while also optimizing for diversity. The latter method utilizes the cosine similarity between the input and the embedded chunks, while penalizing them for closeness to already selected documents (chunks).

5. Experimental setup

5.1. Description

The research presented in this paper employs a three-phase experimental approach. The first phase is exploratory in purpose, with the objective being to define the values of hyperparameters that generate optimal results and to select the LLM that will be used in the second phase (main) of experiments. The second experimental phase aims to systematically evaluate the impact of each RAG component to the role-playing simulation in the OE process, reporting qualitative and quantitative features of the generated ontologies. The third experimental phase is based on the findings of the second one, with the objective of addressing the challenges that are raised in it. This is achieved by proposing different architectures and methods (as described in Section 6.5).

To achieve this goal, the LLM-generated ontologies are compared with a human-generated (reference) ontology tailored to SAR missions (Masa *et al.*, 2022). To ensure that the LLM-generated ontologies are produced based on the same requirements as the human-generated one, the Prompt Engineer input is crafted to include the same ontology requirements, aim, score, and CQs. The methodological approach followed in this work is guided by the following research questions (RQ) (Doumanas *et al.*, 2024; Doumanas *et al.*, 2025):

- RQ 1: Can LLMs automate the OE process? LLMs can generate text based on vast amounts of data, which could enable them to automate several OE tasks. These include the generation of seed ontological structures, suggestions of refinements, and provision of real-time feedback on the quality of the generated ontology.
- RQ 2: Does the proposed multi-component RAG approach (i.e., domain-specific documents, OWL documentation, ReAct guidelines) effectively supports the generation of coherent, well-structured, and expressive ontologies that cover sufficiently (as judged by the human agent/expert) the domain knowledge? This RQ tests the overall efficacy of the proposed approach.
- RQ 3: Can LLMs effectively process and understand large amounts of disparate data? LLMs have shown remarkable performance in understanding and generation of text. This capability could further enhance the OE process as the RAG components feed the LLM with a large amount of information, for example domain-specific data, OWL axioms examples, and think-action traces.
- RQ 4: Are the ontologies generated with the assistance of LLMs of comparable quality regarding accuracy, validity, and expressiveness, to those created solely by human experts? LLMs are trained on extensive datasets and could produce high-quality text (ontological specifications) that meets the criteria for accurate and valid ontology elements. This RQ tests whether LLMs can produce output that human experts deem acceptable.
- RQ 5: Does human involvement remain crucial for refining and validating LLM-generated ontologies, ensuring their overall quality? While LLMs can automate many tasks, they might lack the nuanced understanding required for final refinement and validation. Human experts play a vital role in ensuring that the ontologies are accurate, contextually appropriate, and free from errors and omissions.

RQ 6: Does LLM-powered role-playing simulation within the COE process improve the quality and comprehensiveness of the engineered ontologies? Role-playing could allow agents to adopt different perspectives and better understand the needs and constraints of each role, leading to a more comprehensive ontology. This approach should enhance problem-solving team abilities.

5.2. Evaluated LLMs

LLMs represent the ‘brain’ of the LLM4ACOE framework, as they are responsible for automating the OE process through role-playing simulation. In this version of our framework, we have integrated three of them, that is, ChatGPT4-o, Claude Sonnet, and Gemini Pro, which are commonly accepted as being among the most powerful and widely used models. The framework utilizes the functionality provided by LangChain to make calls to models’ APIs and retrieve the generated responses. However, it should be noted that Prompt Engineers must provide personal credentials, that is, API keys, as well as a sufficient number of available tokens, to utilize these models.

5.3. RAG components

Regarding the domain-specific data used in RAG, ten documents collected from real SAR missions in wildfire incidents were used. The related documents contain information about the elements that we aim to model in the ontology (e.g., missions, number of involved people, involved forces, type of affected area, etc.). It is noteworthy that these documents were initially collected in PDF format but were converted to text files due to LLMs inability to process them effectively (thus, a number of experiments were conducted to solve this problem). During the development of LLM4ACOE it was observed that the ontologies generated using these text files were richer in terms of the number of axioms that is, classes, object properties, etc. compared to those generated using the PDF files.

5.4. Evaluation strategy

In all experimental phases, all the generated ontologies are evaluated against a human-generated SAR ontology (reference ontology) oriented to wildfire incidents, using quantitative measures (i.e., precision, recall, and f1-score) regarding both the classes and object properties. The selection of this domain (SAR) is justified by the fact that it constitutes a dynamic domain, in which a substantial amount of data, coming from sensors, applications, open data, etc., is generated in real-time. Therefore, the appropriate representation of these data can further enhance the vision and the decisions taken by stakeholders. In our previous related work with other domains (e.g., Parkinson’s disease) we have observed a general domain-independency of LLMs performance for OE (Doumanas *et al.*, 2025). Having said that, the focus of this paper is not to evaluate the proposed framework against different LLMs and different domains, but to evaluate the agentic role-playing simulation of COE process, enhanced with different RAG components, towards a fully automated ACOE process.

Moreover, we need to consider that different ontologies model the domain knowledge differently. For example, the knowledge: ‘*An incident involves vulnerable objects*’ can be modeled as follows: (a) ‘*incident*’ as a *class*, ‘*vulnerable object*’ as a *class*, and ‘*involves*’ as an *object property*, or (b) ‘*incident*’ as a *class*, ‘*involves*’ as a *data property*, and the ‘*vulnerable object*’ as a *literal*. Subsequently, while we do evaluate the presence of properties, the matches of properties between the LLM-generated and the reference ontology are independent of whether it is an object property in the former and a data property in the latter, or vice versa, as long as they model the same knowledge, as decided by a human (domain) expert.

Moreover, in all experimental phases, a semantic matching approach is adopted: two concepts from the LLM-generated and the reference ontology match if both model the same domain entities. Furthermore, regarding object properties, for matchings to be identified, they should connect semantically equivalent classes via ontology axioms (e.g., *rdfs:domain*, *rdfs:range*). Consequently, object

Table 1. Values of hyperparameters tested in the first experimental phase

Hyperparameters	Tested values		
Temperature	0.0	0.5	1.0
Search method	Similarity	MMR	
Chunk size	750		
Chunk overlap	0		
Retrieved documents	4		

properties in the LLM-generated ontology that lack a connection to a class are not matched to any of the properties, even if they are lexically equivalent to an object property of the reference ontology.

The number of CQs that the LLM-generated ontology can answer is also highlighted as a key area of our interest. Specifically, the LLM is fed (in the prompt) with 18 CQs in natural language (English), as these were defined during the engineering of the human-generated ontology. The objective is to guide the LLM in modelling the specific domain knowledge in a more effective way. The range of CQs that an ontology can answer reflects the ontology's coverage/scope. Subsequently, to evaluate the coverage of the generated ontologies, we report the number of CQs each ontology can answer. To do this, the CQs are manually converted into SPARQL queries by a human expert (ontology engineer). A well-formed SPARQL query requires the presence of all the ontological concepts involved that is, classes, object properties, and/or data properties. Finally, the CQs answered by the generated ontologies are calculated. It must be noted that in each experimental setup we perform independent experiments and report the average scores.

In addition to the quantitative measurements previously outlined, the generated ontologies are also evaluated through the lens of qualitative features. These features encompass (a) the structure (e.g., the appropriate grouping of related classes under a generic superclass), (b) expressiveness (e.g., the utilization of different types of OWL axioms), (c) the presence of syntactical errors, (d) the reuse of existing ontologies, and (e) ontology consistency. The evaluation is conducted by a team of human experts (ontology engineers and domain experts) and tools (Protégé, Pellet reasoner).

5.5. First experimental phase

In the first experimental phase, experiments are conducted using all the proposed RAG components (i.e., domain-specific documents, OWL documentation, and ReAct guidelines). This phase constitutes an exploration phase in which various combinations of LLMs and hyperparameters are evaluated. The goal of this phase is to determine the LLM to be used and the values of hyperparameters that will be utilized in the second (main) experimental phase.

The LLMs selected for this experimental phase are GPT-4o, Claude Sonnet, and Gemini Pro, which are commonly accepted as being among the most powerful and widely used models. Regarding the hyperparameters, the model's temperature values tested are 0.0, 0.5, and 1.0. Moreover, two RAG retrieval methods are employed: (a) the similarity search, and (b) the MMR method, as outlined in Section 4.4. Regarding the model's temperature, the selected values aim to evaluate how the model behaves under varying degrees of randomness and creativity in its responses. Similarly, the selected retrieval methods aim to evaluate the resulting ontologies based on the variety of the retrieved documents.

Finally, certain hyperparameters, such as the chunk size, chunk overlap, and the number of retrieved documents, are defined after thorough experimentation, also considering the type and the amount of the available data. The values of the hyperparameters employed during this experimental phase are presented in Table 1.

It is therefore evident that the experiments conducted in this phase are 18 in total (one for each combination), as three LLMs, three values of temperature and two retrieval methods are employed in

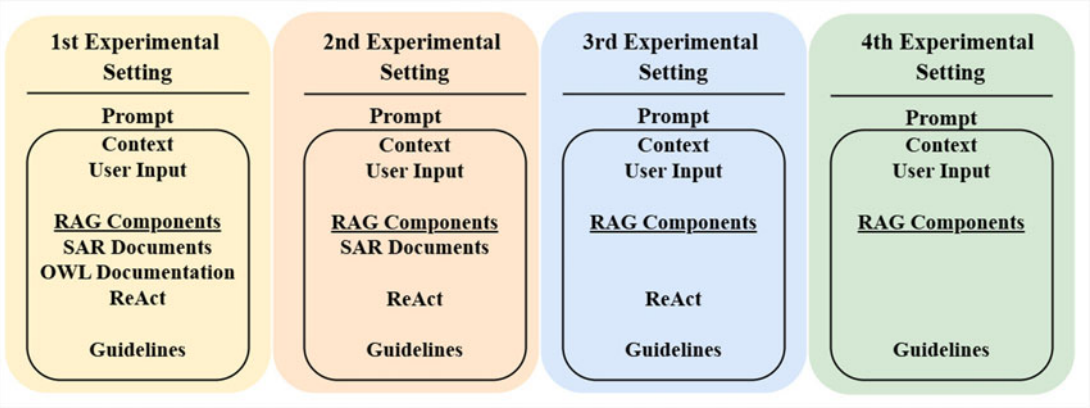


Figure 6. Experimental settings of the second experimental phase

combination. The generated ontologies are then compared with the reference ontology using the evaluation measures (as described in Section 5.4), and the combination of hyperparameters’ values that produces the best results is employed in the following, main experimental phase.

5.6. Second experimental phase

In the second experimental phase, the LLM4ACOE is configured using the best combination of LLM and hyperparameters’ values determined by the first phase to (a) systematically evaluate the contribution of each RAG component in the ACOE process, and (b) evaluate the LLM ability to automate the ACOE process by simulating ACOE roles. To achieve this objective, this experimental phase consists of four experimental settings, as illustrated in Figure 6. In all settings, the same Prompt Engineer input is given to the LLM, while the final prompt varies based on the selected RAG components.

Particularly, in the first experimental setting of phase 2, all the proposed RAG components are used (i.e., SAR documents, OWL documentation, ReAct guidelines). The goal of this phase is twofold: (a) to investigate the capability of the LLM to ‘incorporate’ in the ACOE process all the external information, provided via RAG, enhancing the knowledge of the simulated agents participating in the iterative discussion, (b) to evaluate the contribution of these components in the quality of the generated ontologies.

In the second experimental setting of phase 2, the OWL documentation is removed from the RAG. Thus, the LLM is fed with SAR documents and ReAct guidelines. The goal of this setting is to evaluate whether or to what extent the OWL documentation that is removed from the RAG, contributes to the quality of the generated ontologies.

Similarly, in the third experimental setting of phase 2, the SAR documents are removed from the RAG, and the LLM is only fed with the ReAct guidelines. The goal of this setting is to evaluate the impact of the removed domain-specific documents on the quality of the generated ontologies.

Finally, in the fourth experimental setting of phase 2, the ReAct guidelines are also removed from the RAG, and the only input to the LLM is the prompt (including the context, the Prompt Engineer input, and the general guidelines). The goal of this setting is twofold: (a) to evaluate the impact of the ReAct guideline on the quality of the generated ontologies, and (b) to investigate the ability of the LLM to generate coherent and well-structured ontologies without any further input data.

To address the issue of reproducibility and hallucination that is inherent to LLM responses, ten experiments were conducted for each experimental setting and the average returned values of the measures (precision, recall, f1-score) were calculated. Subsequently, the total number of experiments conducted in the second phase was 40 (10 experiments × 4 experimental settings).

6. Results

6.1. First experimental phase

The objective of the first experimental phase is to assess LLM4ACOE configurations using all three RAG components with various LLMs and hyperparameters' values, as noted above. The LLMs employed in these experiments are ChatGPT4-o, Claude Sonnet, and Gemini Pro.

In the majority of cases, Claude successfully generates well-structured classes. However, it merely identifies the fundamental ones, while leveraging the domain-specific data on a rudimentary level. Furthermore, the resulting ontologies are characterized by the absence of fundamental object and data properties. Moreover, in most cases, ontologies lack basic axioms such as `rdfs:domain` and `rdfs:range`, resulting in incoherent ontologies. This is evident in the measures, as illustrated in Table 2 and in more detail in Table B3 (Appendix B), which demonstrate extremely poor performance, particularly in properties and CQs. Finally, Claude shows a weakness to reuse existing ontologies, failing to adhere to the directives outlined by ReAct guidelines via RAG.

In contrast, Gemini generally succeeds in producing well-structured ontologies, utilizing the domain-specific data available via RAG. The generated ontologies incorporate fundamental object and data properties, but while they are typically well-formed and coherent, they often lack the capacity to adequately model the domain's knowledge. This deficiency is evident in the measures, as evidenced by their notably poor performance, particularly in object properties and CQs. However, Gemini has been observed to successfully reuse existing ontologies in most cases, such as the *opengis*⁷ ontology, adhering to the directives provided by ReAct guidelines via RAG.

Finally, GPT4-o generally succeeds in producing well-structured ontologies, even if it does not fully leverage domain-specific data. Furthermore, it exhibits high performance in object and data properties. The generated ontologies are well-formed and coherent in most cases, successfully modeling most of the domain's knowledge. This is evident in measures which demonstrate high performance, particularly in CQs. However, GPT4-o also exhibits deficiencies in reusing existing ontologies in accordance with the directives provided by ReAct guidelines provided via RAG.

6.2. First experimental phase evaluation

Delving into the results of the first experimental phase, demonstrated in Table 3, the responses of the LLMs vary based on the model and the hyperparameter values. In most cases, the generated ontologies are syntactically sound, with a small number of exceptions where minor errors are detected. After being checked using the Peller reasoner of Protégé, all of them are found to be consistent.

Regarding the domain knowledge captured in resulting ontologies, ChatGPT4-o outperforms the other LLMs, as it is able to leverage the most from the domain-specific data (i.e. SAR documents) taken via RAG. This superiority is further substantiated by the measures, as ChatGPT4-o demonstrates the highest performance across all categories, including the CQs answered by the LLM as it is also reported in Table B3 (Appendix B). Furthermore, ChatGPT4-o is more proficient in terms of the generation of object and data properties, compared to Gemini and Claude that demonstrate significant deficiencies.

Conversely, ChatGPT4-o and Claude Sonnet demonstrate deficiencies in identifying and reusing existing ontologies, even when provided with the requisite instructions or guidelines from ReAct. This is in contrast to Gemini Pro, which effectively reuses existing ontologies. It is noteworthy that, apart from a few cases, none of the selected LLMs have demonstrated success in enriching the generated ontologies with owl axioms beyond the basic ones, while in cases they have achieved this, their number is limited.

Finally, regarding the capability of LLMs to simulate the ACOE process via role-playing, ChatGPT-4o outperforms the other models, exhibiting remarkable capabilities to maintain a productive discussion among the involved roles. Specifically, discussions are concluded by reaching a consensus, among the simulated roles, on the coverage of the resulting ontologies. In contrast, Gemini, and Claude encounter difficulties in sustaining the iterative discussion beyond one or two iterations, completing the iterative

⁷<http://www.opengis.net/ont/swe/2.0>.

Table 2. Comparative results of first experimental phase

LLMs	Classes				Object properties				Properties				Answered CQs (%)
	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	
Claude	16	0.66	0.13	0.20	6	0.10	0.01	0.02	8	0.15	0.01	0.02	0.9
ChatGPT4-o	19	0.67	0.15	0.25	10	0.45	0.08	0.13	22	0.46	0.10	0.17	51.9
Gemini	19	0.44	0.11	0.17	6	0.32	0.03	0.04	11	0.43	0.04	0.08	10.2

Table 3. Results of the first experimental phase

Ontology	Evaluation			Measures						
	Ontology reusability	Consistency (Pellet reasoner)	Syntactical errors	Axioms	Classes	Object properties	Data properties	Subclasses	Equivalent	Disjoint
claude-0.0-750-mmr-4	No	Yes	Minor	40	29	6	2	19	0	0
claude-0.0-750-sim-4	No	Yes	No	43	19	5	0	10	0	0
claude-0.5-750-mmr-4	No	Yes	No	12	7	5	0	1	0	0
claude-0.5-750-sim-4	No	Yes	No	36	16	9	5	10	0	0
claude-1.0-750-mmr-4	No	Yes	No	9	5	0	0	1	0	0
claude-1.0-750-sim-4	No	Yes	No	38	22	9	6	16	0	0
gpt-4o-0.0-750-mmr-4	No	Yes	No	64	17	16	12	22	0	0
gpt-4o-0.0-750-sim-4	No	Yes	No	56	13	13	8	22	0	0
gpt-4o-0.5-750-mmr-4	No	Yes	No	114	27	15	10	12	0	0
gpt-4o-0.5-750-sim-4	No	Yes	No	117	22	6	18	14	1	0
gpt-4o-1.0-750-mmr-4	No	Yes	No	60	19	5	10	8	0	0
gpt-4o-1.0-750-sim-4	No	Yes	No	75	15	7	11	5	0	0
gemini-0.0-750-mmr-4	Yes	Yes	No	85	19	5	6	13	0	0
gemini-0.0-750-sim-4	Yes	Yes	No	113	28	13	12	12	2	0
gemini-0.5-750-mmr-4	Yes	Yes	Minor	56	24	5	0	9	0	0
gemini-0.5-750-sim-4	No	Yes	No	79	24	8	3	12	0	0
gemini-1.0-750-mmr-4	Yes	Yes	No	19	7	1	1	2	0	1
gemini-1.0-750-sim-4	No	Yes	No	47	9	2	10	5	0	0

discussion with sentences such as ‘Conversation continues. . .’, ‘The ontology development process continues iteratively. . .’, or even the discussion stops unexpectedly. Therefore, Gemini and Claude engineer ontologies that fail to adequately capture the domain knowledge. This deficiency is further highlighted by the scores reported.

In terms of the search method employed by RAG, two options are evaluated, with the similarity search demonstrating superiority over the MMR in all reported measures, including CQs, as shown in Table B1 (Appendix B). Conversely, the MMR ensures the diversity of the selected documents, thereby enhancing the overall COE process by providing a more diverse range of information to LLM. Regarding LLMs temperature, as demonstrated in Table B2 (Appendix B), the value of 0.0 produces the optimal results in all measures except for the CQs, where the value of 0.5 yields the best results. However, it is important to note that the temperature value of 0.5 allows the model to deviate from standard responses, thereby enabling the incorporation of further information either from RAG or from inherent knowledge.

Finally, taking into account the combinations of LLMs and hyperparameters, as it is evidenced in Table B3 (Appendix B) that ChatGPT4-o with a temperature value of 0.5 and the MMR retrieval method support/enhance the role-playing agents to generate the best ontology regarding coverage, structure and coherence, reporting superior performance in nearly all measures, including the CQs, where it achieves the highest level of performance, answering 14 out of 18 questions (approximately 78%). Consequently, ChatGPT4-o, with a temperature value of 0.5 and the MMR retrieval method, is selected to be the LLM4ACOE configuration for the second (and main) experimental phase.

6.3. Second experimental phase

The objective of the second experimental phase is twofold: (a) to systematically evaluate the contribution of each RAG component to the LLM-based ACOE process, and (b) to evaluate the LLM ability to automate the ACOE process by simulating role-playing. To strengthen the conclusions drawn, ten iterations were conducted for each experimental setting and the average measures (precision, recall, f1-score) are reported in this section.

6.3.1. First experimental setting of second experimental phase

The results, summarized in Table 4 and presented in Table C2 (Appendix C) in more detail, demonstrate the efficacy of the proposed RAG components since the generated ontologies in most cases manage to capture the basic domain knowledge (that is evident by the CQs measures obtained). This fact contrasts with the low number of axioms (classes, object properties, etc.) the ontologies contain, as the LLM agents in most cases manage to identify only a small portion of them compared to the reference ontology. This is explained by the fact that the basic ontology axioms (i.e., classes, object/data properties) are related to more than one CQ, while the rest of them, which the LLM agents typically fail to incorporate into the engineered ontologies, are related in fewer or even none of the CQs. Finally, this approach generates the most comprehensive ontology regarding the CQs, answering 14 out of 18 CQs.

However, the LLM demonstrates a deficiency in fully exploiting the provided (via RAG components) information, especially the domain-specific information, as can be observed by the total number of classes returned. In contrast, ReAct guidelines appear to ‘steer’ the LLM-simulated Knowledge Engineer agent (from the generated discussions, it is clear that the LLM-simulated agents don’t mix their roles, so we can attribute them with specific capacities) into the correct direction of following the COE process. This is illustrated by the fact that, in half the cases, the LLM attempts to import existing ontologies, as shown in Table C1 (Appendix C). However, in all cases, the imported ontologies are not correctly identified, as the ‘misled’ agent is trying to import the ontology from <http://example.org/otherOntologies/weather.owl> (which constitutes an import example taken from the OWL documentation). On the other hand, in most cases the generated ontologies are well-structured, as the LLM successfully groups relevant classes under a common superclass. Finally, in three

Table 4. Average, SD, and median measures of the first experimental setting

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Average	19.4	0.65	0.16	0.25	10.7	0.41	0.07	0.11	19.5	0.16	0.02	0.04	55
Standard Deviation	7.2	0.06	0.05	0.05	5.25	0.16	0.04	0.07	5.30	0.56	0.11	0.18	15
Median	16.0	0.68	0.14	0.23	9.5	0.42	0.07	0.12	19	0.16	0.02	0.04	58

cases, the LLM manages to incorporate ontology axioms beyond the basic/simple ones, for instance, *owl:inverseOf*, increasing the expressiveness of the generated ontologies.

6.3.2. Second experimental setting of second experimental phase

In the second experimental setting, the OWL documentation is removed from the RAG components, thus the prompt is comprised of the following parts: (a) the context, (b) user input, (c) chunks of domain-specific documents, (d) chunks of ReAct guidelines, and (f) general guidelines. The goal of this setting is to evaluate (by its absence of as a RAG component) the contribution of OWL documentation to the quality of the generated ontologies.

As can be observed in Table C3 (Appendix C), none of the ontologies generated in this setting import other ontologies (not even the experimental ontology, which is imported in some cases in the 1st experimental setting). This is true even though LLM is being fed with this guideline described in ReAct. In terms of ontology structure, approximately half of the ontologies are well-structured, with a significant number of *owl:subclassOf* axioms used to group classes under a relevant superclass. However, there are instances where a common superclass should be applied to a set of related concepts, but the LLM-simulated Knowledge Engineer agent shows a deficiency in identifying these relationships. In the remaining experiments, the LLM fails to correctly organize the classes, indicating suboptimal performance in this field. In the context of expressivity, only three ontologies make use of axioms beyond simple ones, limited to *rdfs:label* and *rdfs:comment*, which are used to describe the ontological concepts. This behavior signifies the beneficial impact of OWL documentation on the generation of expressive ontologies.

Regarding the ontology coverage, as depicted in Table 5 and in more detail in Table C4 (Appendix C), the LLM-simulated agents (Domain Expert, Knowledge Worker) show a remarkable performance in class generation. Subsequently, this approach generates ontologies enriched with a plethora of classes, while there are cases in which necessary classes for SAR missions, such as the *RescueMission*, are present in LLM-generated ontologies but not in the reference ontology. Similar behavior is illustrated for properties, as this approach generates most of them compared to the other experimental settings. It is important to note that in this setting, instances of double properties are observed. For example, a property exists as both an object property and a data property within the same ontology. Furthermore, in these instances, it has been observed that both properties lack either a *rdfs:domain* or *rdfs:range* axioms. Finally, regarding the CQs, even if this approach achieves the highest f1-score in classes and object properties measures (as illustrated in Table 8), the generated ontologies fail to answer the majority of the CQs, achieving performance below 50%. This can be explained since the domain-specific data constitutes a further source of information, coming from real-world scenarios, which were not used during the engineering of reference ontology. Therefore, the resulting ontologies capture and represent domain knowledge that were not represented in the reference ontology, and vice-versa.

6.3.3. Third experimental setting of second experimental phase

In the third experimental setting, the domain-specific documents are removed from RAG components, resulting only to one RAG component present in the experimentation setting, that is, the ReAct guidelines. The prompt consists of the following parts: (a) the context, (b) user input, (c) chunks of ReAct guidelines, and (d) general guidelines. The goal of this setting is to evaluate the contribution of domain-specific information and OWL-specific documentation in the quality of the generated ontologies.

Similarly to the second experimental setting, as reported in Table C5 (Appendix C), none of the generated ontologies import other (external) ontologies, even if the agents are being fed with ReAct guidelines. When it comes to the structure of the ontologies, the LLM agents usually produce well-structured ontologies grouping related classes under a relevant and more abstract superclass. In terms of expressivity, the generated ontologies are pretty basic as the agents do not further enrich them with

Table 5. Average, SD and median measures of the second experimental setting

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Average	34.8	0.19	0.07	0.04	18.3	0.42	0.11	0.16	26.8	0.43	0.10	0.16	43
Standard Deviation	18.14	0.51	0.20	0.30	10.82	0.21	0.05	0.07	9.69	0.18	0.04	0.06	19
Median	29.5	0.19	0.07	0.04	14	0.37	0.11	0.16	26	0.47	0.10	0.17	42

OWL axioms. This finding can be justified by the lack of relevant instructions provided via ReAct to LLM in all iterations as evidenced by the retrieved documents, available to Prompt Engineer through LangSmith. The only axioms present in ontologies, beyond the basic ones, are the *rdfs:comment* and *rdfs:label*, that exist only in three of them.

When it comes to the number of identified classes, the absence of domain-specific information has a negative impact on the number of classes the ontologies contain, as evidenced in Table 6 and in Table C6 (Appendix C). Consequently, in most cases, the LLM agents identify only the classes outlined in the prompt. The sparsity of classes has a detrimental effect on the number of properties identified by the LLM agents, resulting in suboptimal performance. However, the generated ontologies achieve a satisfactory performance by answering approximately half of the queries, thus ranking second among the other approaches. This is because the LLM agents identify the very basic elements/terms of most CQs, while failing to enrich the ontologies with knowledge that should be present in SAR ontology, as happens in the second experimental setting.

6.3.4. Fourth experimental setting of second experimental phase

In this setting, the ReAct guidelines are removed, leaving the LLM agents without any external data, beyond the prompt, to perform the COE process towards the development of a SAR ontology. The goal of this setting is twofold: (a) to evaluate the contribution of the ReAct guidelines in the quality of the generated ontologies and (b) to evaluate the ability of LLM to automatically generate well-structured, expressive and coherent ontologies that capture the knowledge around the SAR missions in wildfire incidents, without further external data, through the role-playing simulation.

Once more, as depicted in Table C7 (Appendix C) none of the generated ontologies reuse existing ones, and they perform similarly to the second and third experimental settings. Regarding the structure, the LLM agents demonstrate a deficiency in generating well-structured ontologies, as they fail to group related classes under an abstract superclass. This behavior demonstrates the positive impact of ReAct guidelines on the ACOE process, as it enhances the knowledge of the Knowledge Engineer with thought-action-observation guidelines about COE tasks. Additionally, the generated ontologies exhibit a limited degree of expressivity, as evidenced by the absence of OWL axioms such as *owl:equivalentWith*, *owl:disjointWith* (and others) in the generated ontologies. This finding is indicative of the positive impact of both OWL documentation and ReAct guidelines on the generation of expressive and well-structured ontologies, as they enhance the knowledge of Knowledge Engineer agent with OWL axioms.

As far as the number of identified classes is concerned, the absence of domain data demonstrates a negative impact on the number of classes contained within the engineered ontologies as illustrated in Table 7 and Table C8, (Appendix C). Consequently, in most cases, the LLM agents identify only the classes outlined in the prompt. A similar trend is observed in the number of properties contained within the ontologies. Furthermore, in one instance, the identified properties are unrelated to any ontology concept. These weaknesses are also reflected in the CQs, as the resulting ontologies fail to answer most of them, reporting the worst performance among all experimental settings, as illustrated in Table 8.

6.4. Results analysis: pros and cons

The results, as depicted in Table 8, demonstrate the effectiveness of the proposed approach, as the RAG components improve the performance of LLM agents in generating more expressive, coherent and complete ontologies, as compared to the agents simulated in the bare LLM configuration. In terms of structure, the combination of OWL documentation and ReAct guidelines ensures well-structured ontologies in most cases, while the SAR documents enrich them with domain data and concepts. At the same time, the domain data guides the Domain Expert and Knowledge Worker agents to further enrich the produced ontologies with concepts coming from the datasets. As can be observed from the results, reducing the number of RAG components has a negative impact on the quality of the generated ontologies. The only exception is observed in the number of classes in the second experimental setting of the second

Table 6. Average, SD and median measures of the third experimental setting

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Average	16	0.66	0.13	0.21	8.3	0.40	0.07	0.11	19.4	0.58	0.11	0.18	50
Standard Deviation	5.85	0.13	0.05	0.07	5.39	0.20	0.06	0.09	4.39	0.06	0.03	0.04	11
Median	16.5	0.68	0.14	0.23	6.5	0.37	0.04	0.07	18.5	0.58	0.10	0.18	50

Table 7. *Average, SD and median measures of the fourth experimental setting*

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Average	14	0.75	0.13	0.22	9.9	0.42	0.07	0.12	19.3	0.50	0.10	0.16	39
Standard deviation	3.13	0.05	0.03	0.04	4.53	0.24	0.05	0.08	4.29	0.23	0.04	0.07	20
Median	14	0.72	0.13	0.22	8.5	0.42	0.08	0.13	19.5	0.54	0.10	0.17	36

Table 8. Comparative average measures among experimental settings

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	
Exp. Setting 1	19	0.65	0.16	0.25	11	0.41	0.07	0.11	20	0.56	0.10	0.17	55
Exp. Setting 2	35	0.60	0.23	0.31	18	0.42	0.11	0.16	27	0.43	0.10	0.16	43
Exp. Setting 3	16	0.66	0.13	0.21	8	0.39	0.05	0.09	19	0.58	0.11	0.18	50
Exp. Setting 4	14	0.75	0.13	0.22	10	0.42	0.07	0.12	19	0.50	0.10	0.16	39

experimental phase, which increases drastically after the removal of the OWL documentation. Another weakness of the approach observed, which is independent of the number of RAG components involved, is the relatively poor performance of the LLM agents in generating object and/or data properties. This deficiency has a significant impact not only on the quality of the generated ontologies, as they lack coherence, but also on the effective representation of the domain knowledge in general. Furthermore, it is demonstrated that an increase in the classes and properties generated does not necessarily result in an increase in the number of CQs that can be answered. This is due to the presence of entities for example, classes and properties, in multiple CQs. The absence of these entities in the generated ontologies has a greater impact than the presence of entities that only appear in a single CQ. Another justification for this observation is the existence of entities that are not present in the reference ontology. Finally, it is notable that the experimental setting integrating all the proposed RAG components (first experimental setting) is the only one that imports and reuses ontologies, even if these ontologies are not real (dummy/imaginary examples), and serve merely as illustrative examples in the OWL documentation.

As demonstrated by the obtained results, each RAG component contributes to the ACOE process in a unique way, enhancing the existing knowledge of the simulated agents derived from the LLM's training data. The combination of all RAG components achieves the best results in terms of the selected evaluation measures, especially for the CQs, while also it outperforms the rest combinations in terms of expressivity and structure. Therefore, the first experimental setting not only produces the best ontology, in terms of the number of CQs answered, but also achieves the highest score in general.

However, this approach is also subject to certain limitations that affect the quality of the resulting ontologies. Some of these limitations are due to the quality of the data, such as domain-irrelevant examples that are present in the OWL documentation, and others are related to the ability of the LLM to effectively process and understand large amounts of diverse data. As the results demonstrate, in the first experimental setting of the second experimental phase, the LLM encounters challenges in extracting concepts from SAR documents due to the increased size of the prompt. Conversely, it demonstrates notable performance in this domain in the second experimental setting, where the prompt is more concise due to the absence of the OWL documentation. Furthermore, the OWL documentation feeds the LLM agents with information that is unrelated to the domain, thereby introducing noise with respect to the domain, which, in certain instances, has a detrimental effect on the quality of the generated ontologies. The most interesting side-effect is the reuse of an unreal ontology, as demonstrated in the first experimental setting. Furthermore, there are cases in which the generated ontologies contain irrelevant domain classes, such as *Father*, *Son*, *Daughter*, which are also present in the OWL documentation as examples of OWL axiom usage. Of course, such dummy/educational examples of OWL axioms may be removed from the OWL documentation prior to its use in the RAG components of the proposed approach.

6.5. Third experimental phase

As previously stated, to address the challenges raised, a third phase of experimentation is carried out. In this phase, two more experiments are conducted using all the proposed RAG components, following different architectures and methods. To enhance the generated results five iterations of experiment are conducted.

6.5.1. Augmentation of domain-specific data

In the first experiment, the objective is to tackle the challenge of restricted domain knowledge derived from domain-specific data via RAG, in the generated ontologies, as demonstrated in the first experimental setting of the second phase. Towards this direction, the chunk size of the SAR documents is doubled in comparison to the other RAG components, providing the LLM with a more substantial amount of domain-specific data.

As illustrated in Table 9, and in more detail in Table D1 (Appendix D), the outcomes of this approach do not align with the anticipated results. In most experiments, the number of classes in the resulting

Table 9. Average, SD and median measures of augmentation of domain-specific data

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Average	40	0,57	0,21	0,28	13,2	0,36	0,08	0,13	27	0,42	0,10	0,16	33
Standard Deviation	35,1	0,20	0,06	0,02	2,79	0,17	0,04	0,06	11,75	0,15	0,03	0,05	12
Median	24	0,54	0,18	0,27	14	0,43	0,08	0,14	23	0,47	0,10	0,13	39

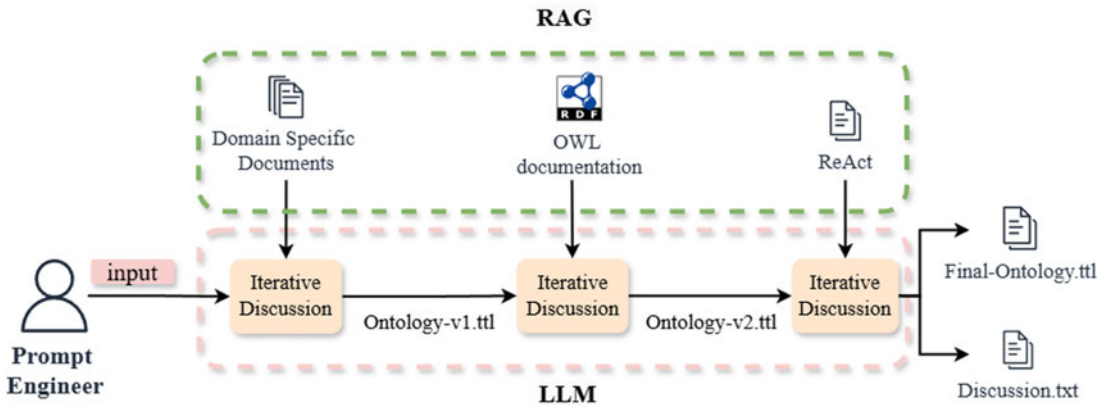


Figure 7. *Sequential Approach*

ontologies remains consistent with the first experimental setting. However, an exception to this observation is noted in the final experiment, where the resulting ontology consists of 110 classes (the higher number observed in the experimental results), a significantly higher number compared to the human-generated ontology, outperforming all the experiments to this point. A similar behavior is observed in object and data properties, as their number does not increase dramatically, demonstrating similar performance with the second experimental phase. The observations are further substantiated by metrics of CQs, which demonstrate the poorest observed performance (as illustrated in Table 10).

As has already been discussed, it is evident that LLM agents demonstrate a weakness to reuse existing ontologies across the five experiments via RAG, as also depicted in Table D2 (Appendix D), despite the provision of ReAct guidelines. Conversely, the generated ontologies in most cases are well-structured, as the LLM agents manage to group them appropriately when necessary, using the related OWL axioms, as described in OWL documentation and ReAct guidelines. However, the generated ontologies exhibit a deficiency in expressivity, as none of them are enriched with related OWL axioms, beyond the basic/simple ones. The only exception is observed in two ontologies, where the first uses two *owl:inverseOf* axioms, while the latter uses *rdfs:label* and *rdfs:comment* axioms to describe the ontology concepts.

6.5.2. Sequential approach

In the second experiment, having recognized the deficiency of the LLM to effectively process and understand large amounts of heterogeneous data, a new approach is adopted. In contrast to the preceding experiments, in which the whole process is performed in parallel, that is the prompt consists of data from all RAG components and the LLM agents exploit these at the same time, in this experiment the COE process follows a sequential approach, in which one RAG component contributes to the ACOE process at a time, enhancing the information provided to the LLM agents, following a pipeline approach, as depicted in Figure 7. The objective of this approach is to fully leverage the additional information towards the development of more expressive, coherent, and well-structured ontologies that capture the domain knowledge demonstrating a high degree of coverage.

In this three-step approach, the initial step involves the retrieval of domain-specific data (SAR documents) and its subsequent provision to the LLM, expanding the knowledge of the Domain Expert and the Knowledge Worker simulated agents. The prompt is comprised of the following components: (a) the context, (b) user input, (c) chunks of domain-specific data, and (d) general guidelines. The initial version of the ontology is then created. In the subsequent step, the OWL documentation RAG component is employed, enriching the inherent knowledge of Knowledge Engineer and Knowledge Worker, to reconstruct and enrich the generated ontology with owl axioms. The prompt comprises the following components: (a) the context, (b) the generated ontology, (c) chunks of OWL documentation, and

(d) general guidelines. Finally, in the third step, the ReAct component is used, enhancing the inherent knowledge of Knowledge Engineer, to further refine the resulted ontology.

Regarding the simulation of roles, in the initial step the three simulated agents are asked to perform an iterative discussion aiming to engineer an ontology with great coverage that captures the most of the domain knowledge leveraging both the domain knowledge of LLMs and the domain-specific information that is available via RAG. In the second step, the simulated roles are prompted to restructure the ontology engineered in the first step, enhancing its expressivity by enriching it with OWL axioms, as described in OWL documentation that is available via RAG. Finally, in the third step, the goal of simulated roles is twofold, (a) to further refine the ontology, engineered in the second phase, leveraging the ReAct guidelines available via RAG and (b) to act as a ‘filter’, removing irrelevant concepts that is, classes and/or properties that may exist in the engineered ontology due to the noise inherent to OWL documentation.

As illustrated in Table 10, and in more detail in Table D3 (Appendix D), the sequential approach outperforms the rest approaches, demonstrating a remarkable performance as regards the class generation, fully exploiting the domain-specific data, coming from the SAR documents via RAG. A similar pattern is also observed in the domain of properties, as the number of object properties are increased by 33% on average compared to the second-best performance achieved in the second experimental setting, as illustrated in Table 10. These outcomes are also reflected in the measures, achieving the highest scores in all of them. Especially for the CQs, the ontologies engineered by the proposed sequential approach, manage to answer at least half of them in all the experiments.

Similarly to the rest of the experiments, except from the first experimental setting, the LLM agents fails to reuse existing ontologies in the produced ontologies, as illustrated in Table D4 (Appendix D), showing a deficiency in executing the guidelines taken by the ReAct via RAG. Conversely, this approach consistently generates well-structured ontologies, largely due to the incorporation of an appropriate number of *owl:subclassOf* axioms as needed, following the ReAct guidelines effectively. Furthermore, the generated ontologies are enriched with additional owl axioms, including *owl:equivalentTo* and *owl:inverseOfObjectProperty*, thereby enhancing their robustness.

7. Overall discussion and limitations of the approach

The results reported in Table 10 demonstrate the efficacy of the proposed RAG-based approach in the COE process. The additional information provided to LLMs via RAG components extends the knowledge of the simulated agents, while it further guides them in generating coherent and well-structured ontologies. These ontologies are not capable of capturing domain-knowledge at an acceptable level, given also that no human intervention beyond the initial input defining the requirements, aim, scope and CQs of the ontology is supported.

The limitations that are due to the capacities of LLM agents and the information provided via RAG components can have a detrimental effect on the quality of the resulting ontologies. To address the challenges raised during the second experimental phase, an additional experiments was conducted using a ‘curriculum’-based approach and framework architecture, where information is provided to LLM agents gradually, following an ontology refinement approach according to OE methodological guidelines.

As demonstrated in Table 10, the augmentation of the chunk size for SAR documents does not have a substantial effect on the quality of the generated ontologies, as the resulting ontologies demonstrate a lack of domain knowledge, as evidenced by the scores of CQs, reporting the poorest performance. Furthermore, except for a single case, this approach does not exert a substantial influence on the number of classes generated, exhibiting also a deficiency in expressivity as lack further OWL axioms beyond the basic ones. This observation lends further support to the hypothesis that the LLM exhibits deficiencies in processing and comprehending voluminous and disparate data.

Conversely the sequential approach adopted in the third experimental phase facilitates the agents to generate ontologies that are more coherent, and well-structured, while performing better on CQs, achieving the highest scores. Regarding the expressivity of the resulting ontologies, the sequential approach

Table 10. *Comparative average measures among all experimental approaches*

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	
Exp. Setting 1	19	0.65	0.16	0.25	11	0.41	0.07	0.11	20	0.56	0.10	0.17	55
Exp. Setting 2	35	0.60	0.23	0.31	18	0.42	0.11	0.16	27	0.43	0.10	0.16	43
Exp. Setting 3	16	0.66	0.13	0.21	8	0.39	0.05	0.09	19	0.58	0.11	0.18	50
Exp. Setting 4	14	0.75	0.13	0.22	10	0.42	0.07	0.12	19	0.50	0.10	0.16	39
Doubled chunk	40	0.57	0.21	0.28	11	0.36	0.08	0.13	27	0.42	0.10	0.16	33
Sequential	44	0.44	0.24	0.31	27	0.42	0.16	0.22	44	0.33	0.14	0.19	59

demonstrates a similar behavior compared to the first experimental setting, in which all the proposed RAG components are used at the same time. Moreover, the agents have demonstrated a capacity to identify and incorporate a greater number of classes into the resulting ontologies, thereby fully leveraging the domain-specific data provided by the RAG components. Furthermore, the generated ontologies are well-structured in most experimental cases, as the agents demonstrate a noteworthy capacity to, not only capture the requisite domain knowledge beyond the fundamental one, but also to structure it correctly. A similar behavior is also observed in the case of properties' generation, as the sequential approach outperforms the rest of the approaches in generating coherent ontologies. The efficacy of this approach is further substantiated by measures, which consistently report the most favorable outcomes, eclipsing all competing approaches.

A key finding of this research is the deficiency of the LLM (in this paper, the ChatGPT4-o) to fully exploit large amounts of disparate data targeting different abilities, concurrently. This weakness is observed during the first experimental setting of the second experimental phase, in which even if the agents are provided with additional information leveraging the three proposed components, they show a deficiency in producing ontologies that capture the domain-knowledge at an acceptable level, generating ontologies with a small number of elements (i.e., classes and object/data properties). A similar pattern is also observed in the third experimental phase, where the domain-specific data are augmented, showing a deficiency not only in the representation of the domain knowledge in an acceptable way, but also in the expressivity and quality of the generated ontologies in general. Conversely, the sequential approach generates better results across all measures, surpassing all other approaches. This approach underscores the distinct contributions of each proposed RAG component. In particular, it is evident that the use of domain-specific data assists LLMs in developing ontologies with increased domain knowledge and greater coverage, thereby addressing the limited domain knowledge observed in some domains, for example, the SAR domain as demonstrated in other works (Doumanas *et al.*, 2025).

In relation to the OWL documentation, it is evident that it serves as an additional guideline for the LLM's performance, since employing examples of OWL axioms enhances the expressivity of the resulting ontologies. The experimental findings demonstrate that the ontologies generated with this RAG component exhibit a higher level of expressivity compared to those generated without it. However, it is important to note that the presence of irrelevant data in the examples may mislead the RAG-enhanced LLM. This phenomenon is also evident in the initial exploration experiments, where the values of certain hyperparameters are being investigated. It is observed that an increased chunk size results in ontologies that are afflicted by the presence of irrelevant concepts. The reduction of the chunk size to 750 words has been demonstrated to be an effective strategy in addressing this challenge, thereby minimizing the number of irrelevant concepts obtained from OWL documentation.

Finally, the ReAct approach can assist the agents in 'thinking' and 'acting' in their role-playing abilities, facilitating the generation of expressive and well-structured ontologies. However, as evidenced by the retained results, ChatGPT4-o exhibits deficiencies in fully adhering to these guidelines, as it lacks the capability to discover (in the Web) and reuse existing ontologies. In contrast, Gemini demonstrates a capability to identify and reuse related ontologies, as evidenced in the initial experimental phase, thereby outperforming other LLMs in this field.

Concerning the LLM-powered role-playing simulation, it is noteworthy that the LLM (ChatGPT-4o), is capable of keeping an iterative and productive discussion among the agents for the engineering of acceptable ontologies outperforming the Claude Sonnet and Gemini Pro. Moreover, the enhancement of agents with additional information (i.e., domain-specific data, OWL documentation, ReAct guidelines) facilitate the development of acceptable ontologies, especially in the sequential configuration, regarding the accuracy, validity, coverage, and expressiveness. However, the results also indicate that the roles (agents), are not disjointed, using the same information provided via RAG. Subsequently, the agents fail to adopt different perspectives and better understand the needs and constraints of each role. The reuse of non-existing ontologies and the incorporation of irrelevant domain classes are illustrative examples that substantiate this claim. In the presence of fully disjoint agents, the incorporation of these irrelevant and erroneous imports would be prevented. The results demonstrate that the proposed LLM4ACOE

framework, especially the sequential configuration of it, has the capacity to automate the ACOE process by facilitating the ontology development surpassing the inherent limitations of conventional approaches, producing in most cases acceptable ontologies, that capture at least the basic domain knowledge, and consequently deliver expressive and well-structured ontologies. However, as demonstrated by the results, the LLM responses (i.e., generated ontologies) are not always consistent among iterations, while in some cases fail to represent all the necessary knowledge or even incorporate irrelevant or misleading knowledge.

It must be noted that the engineering of incoherent and inconsistent ontologies or even the existence of syntactical errors or bias, can have severe consequences. Therefore, considering the criticality of some domains, such as the one encountered in this study, human involvement remains crucial for refining and validating the LLM-generated ontologies during their engineering (not only at the end), ensuring their overall quality. Human experts still play a significant role in ensuring that the ontologies delivered for their integration in applications and systems are accurate, contextually appropriate, and free of biases, errors and omissions. This conclusion is fully consistent with the experimental results/conclusions of our previous research work, namely SimX-HCOME (Doumanas *et al.*, 2025), in which the synergy between Machine (LLM)-Human towards the COE reports similar results/measures.

8. Conclusion

The work presented in this paper extends the Sim-HCOME OE methodology by incorporating additional knowledge to COE simulated agents via the LLM4ACOE framework, integrating three RAG components. Towards this direction, an LLM-based and RAG-enhanced framework is implemented with the aim to evaluate the automation of the engineering domain-specific ontologies via role-playing simulation. The experiments presented in this paper highlight the efficacy of the proposed RAG components in the proposed ACOE process, generating coherent, robust, and well-structured ontologies that capture further domain knowledge, tackling the problem of incomplete domain knowledge that is inherent to the LLMs. The results highlight the ability of the proposed approach, especially the sequential one, to effectively automate the ACOE process, through role-playing simulation, surpassing most of limitations inherent to the LLMs.

The experimental results answer to the research questions (RQ) defined at Section 5.1, as follows: They demonstrate the capacity of LLMs (ChatGPT4o in this case) to automate the COE process through role-playing simulation, engineering in some cases acceptable ontologies by human experts/agents (RQ1). Moreover, the proposed RAG components enhance the LLM agents with additional knowledge supporting/facilitating the engineering of coherent, well-structured, and expressive ontologies that cover the domain knowledge adequately in most cases (RQ2). The results obtained demonstrate a close correlation between the quality of the engineered ontologies and the quality of the external information available to the simulated roles (LLM agents) via RAG. The inclusion of irrelevant concepts in external information results in the introduction of noise into the engineered ontologies. Furthermore, the results demonstrate the deficiency of the LLM (ChatGPT4o) to fully exploit the diverse information provided via RAG (RQ3). The additional experiments that were conducted, augmenting the domain-specific documents, validate the aforementioned claim as the generated ontologies demonstrated a detrimental quality. On the other hand, the sequential approach followed, in which one RAG component contributes to the ACOE process each time, resulted in acceptable ontologies concerning accuracy, validity, coverage and expressiveness (RQ4). However, taking into account the inconsistent responses of LLM amongst iterations, the failure to represent all the necessary knowledge, and the inclusion of irrelevant or misleading knowledge, it is concluded that human involvement remains crucial for refining and validating LLM-generated results (ontologies in our case), ensuring their overall quality (RQ5). The integration of LLM-powered role-playing simulation into the COE process generally improves the quality of the engineered ontologies. However, the experimental results also indicate that the roles (agents) are not disjointed using the same information provided via RAG. As a result, the agents fail to adopt different perspectives and better understand the needs and constraints of each role (RQ6).

Future work includes further experimentation with the proposed sequential approach, evaluating different architectures regarding the sequence of the proposed RAG components. In addition, further experimentation is planned with distinguished agents using disjoint instances of LLMs and RAG that are capable of using tools tailored to OE tasks such as searching the Web for relevant ontologies (to integrate them with those engineered by the LLM), performing validation checks (syntax, consistency), etc. Finally, the integration and evaluation of GraphRAG as well as a fine-tuning LLM tailored to engineer ontologies are left for future work, as both have the potential to further enhance the proposed ACOE process.

Appendix A

In this section the prompt templates used to conduct the experiments and produce the results, are presented.

Prompt Example using the parallel approach:

Create three instances of yourself playing three different roles in the ontology engineering process based on the HCOME collaborative ontology engineering methodology. The three roles are the knowledge engineer, the domain expert and the knowledge worker. These three roles work together to create an ontology. The Knowledge Engineer is responsible for the requirements specification, conceptualization and generation of the ontology. The Domain Expert is an experienced person and provides the requirements for the ontology, terminology, definitions of terms, domain specific explanations of terms and his experience in general. The Knowledge Worker is the user of the ontology and actively participates in the ontology engineering process. The above roles should express their deep knowledge during the conversation. Their aim is to play all three roles, simulating the HCOME methodology. The above mentioned roles will interact with each other, asking and answering questions until a valid and comprehensive ontology is created, which covers all the defined requirements below.

The aim of the ontology to be created is to model all the necessary concepts and their relationships for Search and Rescue (SAR) missions. The scope of the ontology is wildfire incidents. The generated ontology should be able to capture, link and semantically integrate heterogeneous data, regarding the environment in which the mission takes place, collected from different resources such as sensors, social media (from users in the nearby area), and input from first responders, in order to provide decision support services to the crisis management centre. Therefore, the generated ontology should have a deep scope, encompassing a wide range of domain knowledge relevant to forest fire emergencies. The key knowledge that must be represented in your ontology includes (A) Incidents and Impacts: The ontology must capture relevant incidents and impacts in a wildfire disaster. This knowledge is crucial for understanding the extent and severity of wildfire and its consequences. (B) Weather Conditions: Representation of weather conditions, including temperature, wind speed, humidity, and weather forecasts, is essential for understanding the environmental factors influencing the behavior of wildfire. This knowledge helps in assessing the potential spread and behavior of the fire. (C) Data from Human and Earth Observations: The ontology must include data relevant to the analysis of input data coming from various type of sensors, satellites, and social media sources. This knowledge provides valuable information for monitoring and assessing the wildfire situation. (D) Missions and Relationships Between Services: Representation of missions and relationships between the services involved in wildfire management is important for coordinating and organizing emergency response efforts. Moreover, you will be given three sets of competency questions. The competency questions are the following: The first set aims to represent the wildfire disaster and relevant incident and impacts: (CQ1) What are the most important weather variables that can cause forest fire? (CQ2) What are the current measurements for these weather variables? (CQ3) What is the forecast for the weather in this location? (CQ4) Where did the incident take place? (CQ5) What is the priority of an incident during a forest fire disaster? (CQ6) What incidents

during forest fires are the most urgent? Set of CQs that you must take into account related data from human and earth observations: (CQ7) What data from the source is depicted? (CQ8) Which is the creation date of these data? (CQ9) What is the location of this item? (CQ10) Which is the classification type of smoke? (CQ11) Which vulnerable objects were involved in the incident? (CQ12) What is the status of wildfire forestry works (firebreaks, access to forest roads, etc.)? The last set of CQs that your ontology must answer is related to the representation of missions and relationships between the services: (CQ13) What services or support do you offer for firefighting? (CQ14) Which mission do you follow for this support/service? (CQ15) What is the location where this mission is taking place? (CQ16) Where is the most urgent mission taking place? (CQ17) What is the population density in the area? (CQ18) What is the location of the people involved? Your ultimate goal is to generate a comprehensive ontology that covers all of the above requirements and is capable of answering the above questions. You need to create an extensive, comprehensive and well-connected ontology using all the necessary owl axioms to meet all the above requirements.

During the discussion and design of the ontology you should consider the following additional content. You should follow the above way of thinking-acting-observing, but **BEHIND THE SCENES** and **WITHOUT** showing this thinking chain during the discussion and the ontology generation process, as given below

START OF REACT

{react_context}

END OF REACT

You should also use the some of the following OWL axioms to make the ontology as expressive as possible. Use **ONLY THE ONTOLOGY AXIOMS** given in the **EXAMPLES** and **NOT** the **DATA PRESENTED**. You do not need to use all of them, but only the necessary axioms to create a **WELL CONNECTED** and **EXPRESSIVE** ontology.

START OF OWL DOCUMENTATION

{owl_context}

END OF OWL DOCUMENTATION

The following data describe real Search and Rescue (SAR) missions. You should **EXTRACT** related concepts (**CLASSES** and **OBJECT PROPERTIES**) and add them to the generated ontology. Try to extract as much relevant classes and properties as possible.

START OF DOMAIN DATA

{sar_context}

END OF DOMAIN DATA

The iterative discussion stops when the generated ontology answers all the given competency questions and covers all the requirements of the ontology. Thus create as many classes and properties as possible. Feel free to use domain knowledge to extend the ontology with classes and properties to make it as comprehensive as possible. **DO NOT STOP** until cover all the given requirements. Present the iterative discussion and the generated ontology in Turtle (TTL) format **WITHOUT** individuals.

Appendix B

In this section the results of the first experimental phase are presented.

Table B1. Comparative average measures among similarity method of RAG in the first experimental phase

Similarity Method	Classes				Object properties				Properties				Answered CQs (%)
	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	
Similarity	19	0,63	0,14	0,23	8	0,28	0,04	0,07	16	0,39	0,06	0,10	24
MMR	17	0,55	0,12	0,19	6	0,29	0,03	0,06	11	0,31	0,04	0,07	18

Table B2. *Comparative average measures regarding LLM temperature in the first experimental phase*

Temperature	Classes				Object properties				Properties				Answered CQs (%)
	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	Average number	Precision	Recall	F1-score	
0.0	21	0,64	0,16	0,25	10	0,35	0,06	0,10	16	0,48	0,08	0,13	21
0.5	20	0,56	0,13	0,21	8	0,26	0,04	0,06	14	0,29	0,05	0,08	26
1.0	13	0,58	0,10	0,16	4	0,26	0,01	0,02	10	0,28	0,03	0,05	16

Table B3. Comparative average measures of the first experimental phase

Ontology	Classes			Object properties			Properties			Answered CQs (%)
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	
claude-0.0-750-mmr-4	0.52	0.19	0.28	0.33	0.03	0.06	0.38	0.03	0.05	39
claude-0.0-750-sim-4	0.63	0.15	0.24	0.60	0.05	0.09	0.80	0.04	0.07	6
claude-0.5-750-mmr-4	0.71	0.06	0.11	0.40	0.03	0.06	0.60	0.03	0.06	6
claude-0.5-750-sim-4	0.63	0.13	0.21	0.33	0.05	0.09	0.57	0.08	0.14	28
claude-1.0-750-mmr-4	0.80	0.05	0.09	0	0.00	0	0	0.00	0	0
claude-1.0-750-sim-4	0.68	0.19	0.29	0.33	0.05	0.09	0.53	0.08	0.14	44
gpt-4o-0.0-750-mmr-4	0.71	0.15	0.25	0.44	0.12	0.18	0.57	0.16	0.25	39
gpt-4o-0.0-750-sim-4	0.92	0.15	0.26	0.46	0.10	0.16	0.57	0.12	0.20	50
gpt-4o-0.5-750-mmr-4	0.67	0.23	0.34	0.40	0.10	0.16	0.48	0.12	0.19	78
gpt-4o-0.5-750-sim-4	0.55	0.15	0.24	0.83	0.08	0.15	0.58	0.14	0.22	67
gpt-4o-1.0-750-mmr-4	0.58	0.14	0.22	0.40	0.03	0.06	0.80	0.12	0.21	33
gpt-4o-1.0-750-sim-4	0.60	0.11	0.19	0.14	0.02	0.03	0.44	0.08	0.13	44
gemini-0.0-750-mmr-4	0.47	0.11	0.18	0.20	0.02	0.03	0.36	0.04	0.07	6
gemini-0.0-750-sim-4	0.57	0.20	0.30	0.38	0.08	0.14	0.44	0.11	0.17	28
gemni-0.5-750-mmr-4	0.38	0.11	0.17	0.20	0.02	0.03	0.60	0.03	0.06	0
gemini-0.5-750-sim-4	0.42	0.13	0.19	0.13	0.02	0.03	0.09	0.01	0.02	11
gemini-1.0-750-mmr-4	0.14	0.01	0.02	1.00	0.02	0.03	0.50	0.01	0.02	6
gemini-1.0-750-sim-4	0.67	0.08	0.13	0.00	0.00	0	0.58	0.07	0.12	11

Appendix C

In this section the results of the second experimental phase are presented.

Table C1. Experimental results from the first experimental setting

Ontology	Evaluation			Measures						
	Ontology reusability	Consistency (Pellet reasoner)	Syntactical errors	Axioms	Classes	Object properties	Data properties	Subclasses	Equivalent	Disjoint
Experiment 1	Yes	Yes	No	118	15	7	8	15	1	0
Experiment 2	No	Yes	No	57	16	5	14	19	19	2
Experiment 3	Yes	Yes	No	52	16	5	6	11	3	0
Experiment 4	Yes	Yes	No	75	15	9	10	19	6	1
Experiment 5	No	Yes	No	97	16	8	15	23	3	0
Experiment 6	Yes	Yes	No	92	22	10	8	18	17	0
Experiment 7	No	Yes	No	68	23	10	5	15	0	0
Experiment 8	No	Yes	No	61	13	22	9	31	17	0
Experiment 9	Yes	Yes	No	59	19	13	7	20	20	1
Experiment 10	No	Yes	No	111	39	18	6	24	0	0

Table C2. Measures of the first experimental setting

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Experiment 1	15	0,73	0,14	0,23	7	0,57	0,07	0,12	15	0,80	0,12	0,21	56
Experiment 2	16	0,63	0,13	0,21	5	0,40	0,03	0,06	19	0,58	0,11	0,18	39
Experiment 3	16	0,69	0,14	0,23	5	0,40	0,03	0,06	11	0,82	0,09	0,16	22
Experiment 4	15	0,67	0,13	0,21	9	0,44	0,07	0,12	19	0,63	0,12	0,20	61
Experiment 5	16	0,69	0,14	0,23	8	0,50	0,07	0,12	23	0,52	0,12	0,19	78
Experiment 6	22	0,73	0,20	0,31	10	0,40	0,07	0,11	18	0,50	0,09	0,15	67
Experiment 7	23	0,57	0,16	0,25	10	0,50	0,08	0,14	15	0,53	0,08	0,14	56
Experiment 8	13	0,69	0,11	0,19	22	0,18	0,07	0,10	31	0,39	0,12	0,18	61
Experiment 9	19	0,58	0,14	0,22	13	0,08	0,02	0,03	20	0,25	0,05	0,08	44
Experiment 10	39	0,56	0,28	0,37	18	0,61	0,18	0,28	24	0,58	0,14	0,22	67

Table C3. *Experimental results from the second experimental setting*

Ontology	Evaluation			Measures						
	Ontology reusability	Consistency (Pellet reasoner)	Syntactical errors	Axioms	Classes	Object properties	Data properties	Subclasses	Equivalent	Disjoint
Experiment 1	No	Yes	No	116	14	13	3	16	0	0
Experiment 2	No	Yes	No	156	65	12	3	15	38	0
Experiment 3	No	Yes	No	160	48	36	1	37	0	0
Experiment 4	No	Yes	No	137	29	15	13	28	12	0
Experiment 5	No	Yes	No	149	22	14	7	21	13	0
Experiment 6	No	Yes	No	155	61	6	11	17	48	0
Experiment 7	No	Yes	No	196	47	42	4	46	11	0
Experiment 8	No	Yes	No	96	18	18	11	29	0	0
Experiment 9	No	Yes	No	162	14	14	10	24	0	0
Experiment 10	No	Yes	No	135	30	13	22	35	0	0

Table C4. Measures of the second experimental setting

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Experiment 1	14	0,93	0,16	0,28	13	0,54	0,12	0,19	16	0,50	0,08	0,14	50
Experiment 2	65	0,43	0,35	0,39	12	0,58	0,12	0,19	15	0,67	0,10	0,17	67
Experiment 3	48	0,46	0,28	0,34	36	0,22	0,13	0,17	37	0,24	0,09	0,13	24
Experiment 4	29	0,52	0,19	0,28	15	0,40	0,10	0,16	28	0,50	0,14	0,22	50
Experiment 5	22	0,77	0,21	0,33	14	0,07	0,02	0,03	21	0,10	0,02	0,03	10
Experiment 6	61	0,41	0,31	0,35	6	0,67	0,07	0,12	17	0,59	0,10	0,17	59
Experiment 7	47	0,43	0,25	0,31	42	0,26	0,18	0,22	46	0,33	0,15	0,20	33
Experiment 8	18	0,72	0,16	0,27	18	0,33	0,10	0,15	29	0,45	0,13	0,20	45
Experiment 9	14	0,86	0,15	0,26	14	0,79	0,18	0,30	24	0,63	0,15	0,24	63
Experiment 10	30	0,50	0,19	0,27	13	0,31	0,07	0,11	35	0,58	0,14	0,22	29

Table C5. Experimental results from the third experimental setting

Ontology	Evaluation			Measures						
	Ontology reusability	Consistency (Pellet reasoner)	Syntactical errors	Axioms	Classes	Object properties	Data properties	Subclasses	Equivalent	Disjoint
Experiment 1	No	Yes	No	100	19	11	13	24	10	0
Experiment 2	No	Yes	No	53	8	5	10	15	0	0
Experiment 3	No	Yes	No	132	14	12	6	18	0	0
Experiment 4	No	Yes	No	102	7	3	12	15	0	0
Experiment 5	No	Yes	No	57	11	1	19	20	6	0
Experiment 6	No	Yes	No	84	19	7	12	19	9	0
Experiment 7	No	Yes	No	120	18	20	10	30	3	0
Experiment 8	No	Yes	No	101	24	6	14	20	18	0
Experiment 9	No	Yes	No	158	25	13	3	16	15	0
Experiment 10	No	Yes	No	69	15	5	12	17	3	0

Table C6. Measures of the third experimental setting

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Experiment 1	19	0,68	0,16	0,26	11	0,64	0,12	0,20	24	0,50	0,12	0,19	61
Experiment 2	8	0,88	0,09	0,16	5	0,40	0,03	0,06	15	0,60	0,09	0,15	50
Experiment 3	14	0,36	0,06	0,11	12	0,25	0,05	0,08	18	0,56	0,10	0,17	56
Experiment 4	7	0,71	0,06	0,11	3	0,33	0,02	0,03	15	0,67	0,10	0,17	50
Experiment 5	11	0,64	0,09	0,15	1	0,00	0,00	0,00	20	0,60	0,12	0,20	39
Experiment 6	19	0,68	0,16	0,26	7	0,71	0,08	0,15	19	0,68	0,13	0,21	50
Experiment 7	18	0,78	0,18	0,29	20	0,65	0,22	0,33	30	0,60	0,18	0,27	67
Experiment 8	24	0,71	0,21	0,33	6	0,33	0,03	0,06	20	0,55	0,11	0,18	39
Experiment 9	25	0,52	0,16	0,25	13	0,31	0,07	0,11	16	0,50	0,08	0,14	61
Experiment 10	15	0,67	0,13	0,21	5	0,40	0,03	0,06	17	0,53	0,09	0,15	28

Table C7. Experimental results from the fourth experimental setting

Ontology	Evaluation			Measures						
	Ontology reusability	Consistency (Pellet reasoner)	Syntactical errors	Axioms	Classes	Object properties	Data properties	Subclasses	Equivalent	Disjoint
Experiment 1	No	Yes	No	54	14	5	8	13	7	0
Experiment 2	No	Yes	No	84	17	19	6	25	0	0
Experiment 3	No	Yes	Minor	58	13	9	6	15	0	0
Experiment 4	No	Yes	No	115	17	16	12	28	3	0
Experiment 5	No	Yes	No	64	18	8	8	16	0	0
Experiment 6	No	Yes	No	56	14	13	7	20	0	0
Experiment 7	No	Yes	No	74	11	10	9	19	0	0
Experiment 8	No	Yes	No	84	17	8	12	20	6	0
Experiment 9	No	Yes	No	74	8	4	16	20	0	0
Experiment 10	No	Yes	No	63	11	7	10	17	0	0

Table C8. Measures of the fourth experimental setting

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Experiment 1	14	0,71	0,13	0,21	5	0,00	0,00	0,00	13	0,00	0,00	0,00	0
Experiment 2	17	0,71	0,15	0,25	19	0,32	0,10	0,15	25	0,32	0,08	0,13	28
Experiment 3	13	0,85	0,14	0,24	9	0,67	0,10	0,17	15	0,60	0,09	0,15	22
Experiment 4	17	0,76	0,16	0,27	16	0,44	0,12	0,18	28	0,39	0,11	0,17	67
Experiment 5	18	0,72	0,16	0,27	8	0,50	0,07	0,12	16	0,69	0,11	0,19	33
Experiment 6	14	0,71	0,13	0,21	13	0,77	0,17	0,27	20	0,80	0,16	0,26	61
Experiment 7	11	0,82	0,11	0,20	10	0,40	0,07	0,11	19	0,53	0,10	0,17	39
Experiment 8	17	0,71	0,15	0,25	8	0,13	0,02	0,03	20	0,40	0,08	0,13	61
Experiment 9	8	0,75	0,08	0,14	4	0,25	0,02	0,03	20	0,55	0,11	0,18	28
Experiment 10	11	0,73	0,10	0,18	7	0,71	0,08	0,15	17	0,76	0,13	0,22	50

Appendix D

In this section the results of the third experimental phase are presented.

Table D1. Measures of the domain-specific data augmentation experiment

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Experiment 1	19	0,84	0,20	0,32	10	0,50	0,08	0,14	21	0,52	0,11	0,18	33
Experiment 2	21	0,67	0,18	0,28	17	0,24	0,07	0,10	23	0,30	0,07	0,11	11
Experiment 3	24	0,54	0,16	0,25	10	0,10	0,02	0,03	17	0,47	0,08	0,13	39
Experiment 4	26	0,54	0,18	0,26	14	0,43	0,10	0,16	24	0,63	0,15	0,24	44
Experiment 5	110	0,24	0,33	0,27	15	0,53	0,13	0,21	50	0,20	0,10	0,13	39

Table D2. Experimental results from the augmentation of domain-specific data

Ontology	Evaluation			Measures						
	Ontology reusability	Consistency (Pellet reasoner)	Syntactical errors	Axioms	Classes	Object properties	Data properties	Subclasses	Equivalent	Disjoint
Experiment 1	No	Yes	No	86	19	10	11	4	0	0
Experiment 2	No	Yes	No	96	21	17	6	0	0	0
Experiment 3	No	Yes	No	171	24	10	7	7	0	0
Experiment 4	No	Yes	No	104	26	14	10	13	0	0
Experiment 5	No	Yes	No	284	110	15	35	72	0	0

Table D3. Measures of the sequential approach

Measures	Classes				Object properties				Properties				Answered CQs (%)
	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	Total number	Precision	Recall	F1-score	
Experiment 1	34	0,44	0,19	0,26	25	0,52	0,22	0,31	51	0,29	0,15	0,20	56
Experiment 2	27	0,44	0,15	0,22	16	0,75	0,20	0,32	41	0,39	0,16	0,22	56
Experiment 3	53	0,43	0,29	0,35	18	0,39	0,12	0,18	32	0,38	0,12	0,18	78
Experiment 4	45	0,49	0,28	0,35	22	0,27	0,10	0,15	30	0,40	0,12	0,18	56
Experiment 5	62	0,40	0,31	0,35	54	0,17	0,15	0,16	65	0,22	0,14	0,17	50

Table D4. Experimental results from the sequential approach

Ontology	Evaluation			Measures						
	Ontology reusability	Consistency (Pellet reasoner)	Syntactical errors	Axioms	Classes	Object properties	Data properties	Subclasses	Equivalent	Disjoint
Experiment 1	No	Yes	No	195	34	25	26	23	1	0
Experiment 2	No	Yes	No	158	27	16	25	16	3	0
Experiment 3	No	Yes	No	144	53	18	14	36	10	0
Experiment 4	No	Yes	No	144	45	22	8	6	1	0
Experiment 5	No	Yes	No	292	62	54	11	29	0	0

References

- Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J., Rytting, C. M. & Wingate, D. 2022. Out of one, many: Using language models to simulate human samples. <https://doi.org/10.48550/arXiv.2209.06899>
- Avila, C. V. S., Vidal, V. M. P., Franco, W. & Casanova, M. A. 2024. Experiments with text-to-sparql based on chatgpt. In *18th IEEE International Conference on Semantic Computing, ICSC 2024, Laguna Hills, CA, USA, February 5–7, 2024*, 277–284. IEEE. <https://doi.org/10.1109/ICSC59802.2024.00050>
- Chang, K. K., Cramer, M., Soni, S. & Bamman, D. 2023. Speak, memory: An archaeology of books known to chatgpt/gpt-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6–10, 2023*, Bouamor, H., Pino, J. & Bali, K. (eds), 7312–7327. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.emnlp-main.453>
- Chen, N., Deng, Y. & Li, J. 2024. The oscars of AI theater: A survey on role-playing with language models. <https://doi.org/10.48550/arXiv.2407.11484>
- DeBellis, M., Duttal, N., Ginoc, J. & Balajid, A. 2024. Integrating ontologies and large language models to implement retrieval augmented generation (rag). *Applied Ontology* **1**, 1–5.
- Doumanas, D., Bouchouras, G., Soularidis, A., Kotis, K. & Vouros, G. 2025. From human- to llm-centered collaborative ontology engineering. *Applied Ontology*. <https://doi.org/10.1177/15705838241305067>
- Doumanas, D., Soularidis, A., Kotis, K. & Vouros, G. A. 2024. Integrating llms in the engineering of a SAR ontology. In *Artificial Intelligence Applications and Innovations - 20th IFIP WG 12.5 International Conference, AIAI 2024, Corfu, Greece, June 27–30, 2024, Proceedings, Part IV*, Maglogiannis, I., Iliadis, L. S., MacIntyre, J., Avlonitis, M. & Papaleonidas, A. (eds), IFIP Advances in Information and Communication Technology 714, 360–374. Springer. https://doi.org/10.1007/978-3-031-63223-5_27
- Fathallah, N., Das, A., De Giorgis, S., Poltronieri, A., Haase, P. & Kovriguina, L. 2024. Neon-gpt: a large language model-powered pipeline for ontology learning. In *Extended Semantic Web Conference, ESWC2024, Hersonissos, Greece*.
- Filippas, A., Horton, J. J. & Manning, B. S. 2024. Large language models as simulated economic agents: What can we learn from homo silicus?. In *Proceedings of the 25th ACM Conference on Economics and Computation, EC 2024, New Haven, CT, USA, July 8–11, 2024*, Bergemann, D., Kleinberg, R. & Sabán, D. (eds), 614–615. ACM. <https://doi.org/10.1145/3670865.3673513>
- Glimm, B., Horrocks, I., Motik, B., Stoilos, G. & Wang, Z. 2014. Hermit: An owl 2 reasoner. *Journal of Automated Reasoning* **53**, 245–269.
- Gui, G. & Toubia, O. 2023. The challenge of using llms to simulate human behavior: A causal inference perspective. <https://doi.org/10.48550/arXiv.2312.15524>
- Hu, Z., Feng, Y., Luu, A. T., Hooi, B. & Lipani, A. 2023. Unlocking the potential of user feedback: Leveraging large language model as user simulators to enhance dialogue system. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21–25, 2023*, Frommholz, I., Hopfgartner, F., Lee, M., Oakes, M., Lalmas, M., Zhang, M. & Santos, R. L. T. (eds), 3953–3957. ACM. <https://doi.org/10.1145/3583780.3615220>
- Huang, K., Meng, X., Zhang, J., Liu, Y., Wang, W., Li, S. & Zhang, Y. 2023. An empirical study on fine-tuning large language models of code for automated program repair. In *38th IEEE/ACM International Conference on Automated Software Engineering, ASE 2023, Luxembourg, September 11–15, 2023*, 1162–1174. IEEE. <https://doi.org/10.1109/ASE56229.2023.00181>
- Kommineni, V. K., König-Ries, B. & Samuel, S. 2024. From human experts to machines: An LLM supported approach to ontology and knowledge graph construction. <https://doi.org/10.48550/arXiv.2403.08345>
- Kosinski, M. 2024. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences* **121**(45), e2405460121.
- Kotis, K. & Vouros, G. A. 2006. Human-centered ontology engineering: The HCOME methodology. *Knowledge and Information Systems* **10**(1), 109–131. <https://doi.org/10.1007/s10115-005-0227-4>
- Litaina, T., Soularidis, A., Bouchouras, G., Kotis, K. & Kavakli, E. 2024. Towards llm-based semantic analysis of historical legal documents. In *SemDH@ESWC*. <https://ceur-ws.org/Vol-3724/short2.pdf>
- Liu, Y., Yao, Y., Ton, J., Zhang, X., Guo, R., Cheng, H., Klochkov, Y., Taufiq, M. F. & Li, H. 2023. Trustworthy llms: A survey and guideline for evaluating large language models' alignment. <https://doi.org/10.48550/arXiv.2308.05374>
- Lo, A., Jiang, A. Q., Li, W. & Jamnik, M. 2024. End-to-end ontology learning with large language models. <https://doi.org/10.48550/arXiv.2410.23584>
- Masa, P., Meditskos, G., Kintzios, S., Vrochidis, S. & Kompatsiaris, I. 2022. Ontology-based modelling and reasoning for forest fire emergencies in resilient societies. In *SETN 2022: 12th Hellenic Conference on Artificial Intelligence, Corfu, Greece, September 7–9, 2022*, 24:1–24:9. ACM. <https://doi.org/10.1145/3549737.3549765>
- Mateiu, P. & Groza, A. 2023. Ontology engineering with large language models. In *25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2023, Nancy, France, September 11–14, 2023*, 226–229. IEEE. <https://doi.org/10.1109/SYNASC61333.2023.00038>
- Pan, X., van Ossenbruggen, J., de Boer, V. & Huang, Z. 2024. A RAG approach for generating competency questions in ontology engineering. <https://doi.org/10.48550/arXiv.2409.08820>
- Papadidis, E. & Kotis, K. 2021. Towards engineering fair ontologies: Unbiasing a surveillance ontology. In *2021 IEEE International Conference on Progress in Informatics and Computing (PIC)*, 226–231. IEEE.
- Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P. & Bernstein, M. S. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023–1 November 2023*, Follmer, S., Han, J., Steimle, J. & Riche, N. H. (eds), 2:1–2:22. ACM. <https://doi.org/10.1145/3586183.3606763>

- Perkovic, G., Drobnjak, A. & Boticki, I. 2024. Hallucinations in llms: Understanding and addressing challenges. In *47th MIPRO ICT and Electronics Convention, MIPRO 2024, Opatija, Croatia, May 20–24, 2024*, Babic, S., Car, Z., Cicin-Sain, M., Ciscic, D., Ergovic, P., Grbac, T. G., Gradisnik, V., Gros, S., Jokic, A., Jovic, A., Jurekovic, D., Katulic, T., Koricic, M., Mornar, V., Petrovic, J., Skala, K., Skvorc, D., Sruk, V., Svaco, M., Tijan, E., Vrcek, N. & Vrdoljak, B. (eds), 2084–2088. IEEE. <https://doi.org/10.1109/MIPRO60963.2024.10569238>
- Poveda-Villalón, M., Gómez-Pérez, A. & Suárez-Figueroa, M. C. 2014. Oops! (ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web & Information Systems* **10**(2), 7–34. <https://doi.org/10.4018/ijswis.2014040102>
- Reddy, G. P., Pavan Kumar, Y. V. & Prakash, K. P. 2024. Hallucinations in large language models (llms). In *2024 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 1–6.
- Salemi, A., Mysore, S., Bendersky, M. & Zamani, H. 2024. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11–16, 2024*, Ku, L., Martins, A. & Srikumar, V. (eds), 7370–7392. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.acl-long.399>
- Shanahan, M., McDonell, K. & Reynolds, L. 2023. Role play with large language models. *Nature* **623**(7987), 493–498. <https://doi.org/10.1038/s41586-023-06647-8>
- Soularidis, A., Kotis, K., Lamolle, M., Mejdoul, Z., Lortal, G. & Vouros, G. 2024. Llm-assisted generation of swrl rules from natural language. In *2024 International Conference on AI x Data and Knowledge Engineering (AIDKE)*, 7–12.
- Updyke, D., Podnar, T. & Huff, S. 2023. Simulating realistic human activity using large language model directives.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R. & Cao, Y. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023*. OpenReview.net. https://openreview.net/forum?id=WE_vluYUL-X
- Zhang, B., Carriero, V. A., Schreiberhuber, K., Tsaneva, S., González, L. S., Kim, J. & de Berardinis, J. 2024. Ontochat: A framework for conversational ontology engineering using language models. <https://doi.org/10.48550/arXiv.2403.05921>
- Zhao, C., Agrawal, G., Kumarage, T., Tan, Z., Deng, Y., Chen, Y. & Liu, H. 2024. Ontology-aware RAG for improved question-answering in cybersecurity education. <https://doi.org/10.48550/arXiv.2412.14191>