Data model for multisensory representations of cultural assets

Document Information

Project Name:	Multisensory, User-centred, Shared cultural Experiences through Interactive Technologies
Project Acronym:	MuseIT
Grant Agreement No:	101061441
Deliverable No:	D3.1
Deliverable Name:	Data model for multisensory representations of cultural assets
Work Package:	WP3
Task:	T3.1
Dissemination Level:	PU
Deliverable Type:	R
Lead Organization:	KCL
Lead Member:	Albert Meroño
Submission month:	M24
Date:	30 September 2024

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Document history

Revision	Date	Description / Reason of change	Submission by
v0.1	05.08.2024	Structure proposal and initial draft	Albert Meroño
v0.2	30.08.2024	First draft for internal review	Nitisha Jain
v0.3	12.09.2024	Second draft addressing review comments	Nitisha Jain, Albert Meroño
v0.4	23.09.2024	Final draft addressing the PMB review comments	Nitisha Jain, Albert Meroño
v1.0	30.09.2024	Final draft submitted to the EU	Renata Sadula

Authors

Partner	Name(s)
KCL	Albert Meroño Peñuela
KCL	Nitisha Jain
KCL	Yihang Zhao
KCL	Bohui Zhang
KCL	Johanna Walker

Contributors

Partner	Contribution type	Name
HB	Review	Thomas v. Erven
CTL	Review	Maria Iosif
CERTH	Review	Panagiotis Petrantonakis

Glossary

Acronym	Definition
CQ	Competency Question
KG	Knowledge Graph
LLMKE	Large Language Models for Knowledge Engineering
LLM	Large Language Model
MultiMO	Multisensory Ontology
OE	Ontology Engineering

Table of contents

Executive Summary 1

1. Introduction 2

1.1 Ontology Engineering: Foundations and Challenges 2
1.2 Competency Questions: Guiding Ontology Design 2
1.3 The Rise of Large Language Models in Ontology Engineering 2
1.4 Entity Linking: Connecting Concepts Across Datasets 3
1.5 Ontology Development with Human-in-the-Loop AI 3
1.6 The Multimodal and Multisensory Ontology (MultiMO) 3

2. Towards Automated Ontology Engineering with Human-in-the-loop 5

2.1 The Promise and Challenge of Large Language Models for Knowledge Engineering 5
2.2 Large Language Models and Knowledge Engineering –State of the Art 5
2.3 Findings 6
2.5 Methodology 7
2.4 Summary 8

3. OntoChat: Supporting User Stories and Competency Questions with Conversation Ontology Engineering 9

3.1 Function 1: Assisted persona and story creation 9
3.2 Function 2: Competency question extraction 10
3.3 Function 3: Competency question filtration and analysis 10
3.4 Function 4: Ontology testing support 11
3.5 Evaluation 11
3.6 Conclusion 117

4. LLMKE: Using Large Language Models for Knowledge Engineering 18

4.1 Introduction 18
4.2 Methods 18
4.3 Results 20
4.4 Conclusion 23

5. MultiMO: An Ontology for Documenting Multimodal and Multisensory Knowledge Graphs 24

5.1 Introduction 24
5.2 Related Work 25
5.3 MultiMO: Documenting Sensory Modalities in Knowledge Graphs 25
5.4 Preliminary Evaluation 29
5.5 Conclusion 31

5. Overall Conclusions 32

References 32

Executive Summary

In the ever-evolving landscape of knowledge representation, ontologies serve as crucial frameworks for structuring and accessing information across various domains. Within the MuseIT project, the need for robust ontologies is particularly pressing, as they play a vital role in the annotation of multisensory and multimodal datasets by providing a standardized vocabulary to describe concepts across these modalities and defining the semantics to help bridge the gap between different types of sensory data. These ontologies are not merely technical tools; they are essential enablers of knowledge access and equity, ensuring that diverse types of sensory and cultural information can be effectively organized, retrieved, and utilized.

However, the field of ontology engineering is currently undergoing a significant transformation. This shift is largely driven by the integration of generative AI techniques into the ontology development lifecycle, turning what was once a highly manual and specialized activity into a semi-automated process that incorporates human expertise at key stages. This human-in-the-loop approach leverages the power of AI to streamline tasks such as entity linking and competency question extraction, while still relying on human judgement to guide and refine the outcomes.

This report focuses on two innovative approaches that exemplify this transformation: LLMKE (Large Language Models for Knowledge Engineering) and OntoChat. These approaches specifically address the challenges of entity linking and competency question extraction, which are critical tasks in the ontology engineering process. By exploring these new techniques, we aim to push the boundaries of how ontologies can be developed and applied, particularly in the context of the novel Multimodal and Multisensory Ontology (MultiMO).

The MultiMO ontology is designed to address the unique challenges of annotating datasets that encompass multiple sensory modalities and cultural contexts. To evaluate the effectiveness of MultiMO, we propose a two-pronged approach. First, we will apply MultiMO to annotate a subset of Wikidata that includes multimodal cultural heritage items, providing a practical test of its capabilities. Second, we will conduct an ethnographic study to understand how ontology engineers interact with and perceive the use of generative AI tools in their work. This study will offer valuable insights into the evolving role of AI in ontology engineering and its potential to enhance or disrupt established practices.

Through this deliverable, we aim to document and critically assess these new methodologies, offering both theoretical and practical contributions to the field of ontology engineering in the age of AI.

1. Introduction

Ontology engineering (OE) has become a cornerstone of knowledge management, enabling the structured representation of information across diverse domains. As the complexity and volume of data continue to grow, particularly in fields involving multisensory and multimodal datasets, the need for advanced ontology engineering techniques has never been more critical. This introduction provides an overview of the key concepts and technologies that are transforming ontology engineering, with a specific focus on the MuseIT project's goals of enhancing knowledge access and equity through the development of the Multimodal and Multisensory Ontology (MultiMO).

1.1 Ontology Engineering: Foundations and Challenges

Ontology engineering is the discipline concerned with the design, development, and maintenance of ontologies, which are formal representations of a set of concepts within a domain and the relationships between those concepts. Traditionally, ontology engineering has been a highly manual and knowledge-intensive process, requiring deep expertise in both the domain of interest and the formal languages used to describe ontological structures.

The primary challenges in ontology engineering include ensuring consistency and completeness, managing the complexity of large-scale ontologies, and facilitating interoperability between different ontological systems. In the context of multisensory and multimodal datasets, these challenges are amplified due to the need to represent diverse types of data—ranging from visual and auditory to tactile and olfactory—within a coherent and accessible framework.

1.2 Competency Questions: Guiding Ontology Design

Competency questions (CQs) are a central concept in ontology engineering, serving as the driving force behind the design and evaluation of an ontology. CQs are essentially queries that the ontology must be able to answer, reflecting the information needs of its intended users. They guide the ontology development process by defining the scope and granularity of the ontology, ensuring that it meets the specific requirements of the domain.

In the context of the MuseIT project, CQs play a critical role in shaping the MultiMO ontology, as they help to identify the key concepts and relationships that must be captured to support the annotation of multisensory and multimodal datasets. The ability to accurately extract and formalize CQs is essential for the ontology's success.

1.3 The Rise of Large Language Models in Ontology Engineering

Recent advancements in artificial intelligence, particularly in the development of large language models (LLMs), have begun to reshape the ontology engineering landscape. LLMs, such as GPT (Generative Pre-trained Transformer) models, can process and generate human-like text based on vast amounts of data. These models have demonstrated remarkable potential in automating various aspects of ontology engineering, including entity linking, knowledge extraction, and CQ generation.

The integration of LLMs into the ontology development lifecycle marks a significant shift from traditional, manual approaches to a more automated and less human-dependent process. This transformation allows

for more efficient handling of large and complex datasets, enabling the creation of more sophisticated and dynamic ontologies. However, the use of LLMs also introduces new challenges, such as ensuring the accuracy and reliability of the generated content and addressing the ethical implications of AI-driven ontology development.

1.4 Entity Linking: Connecting Concepts Across Datasets

Entity linking is a crucial task in ontology engineering, involving the identification and connection of entities (e.g., people, places, concepts) across different datasets. This process is fundamental for building ontologies that can effectively integrate and represent information from diverse sources. In traditional ontology engineering, entity linking was often performed manually, requiring extensive domain knowledge and careful validation.

With the advent of LLMs and other AI techniques, entity linking has become increasingly automated, allowing for more rapid and scalable ontology development. The LLMKE (Large Language Models for Knowledge Engineering) approach, explored in this deliverable, exemplifies how AI can be leveraged to enhance entity linking by extracting relevant entities from large text corpora and linking them to existing ontologies. This approach was the winner of track 2 of the ISWC 2023 LM-KBC Challenge.

1.5 Ontology Development with Human-in-the-Loop AI

Despite the advances in AI-driven ontology engineering, human expertise remains indispensable. The concept of human-in-the-loop AI refers to systems where AI and human experts collaborate, with AI handling routine or large-scale tasks and humans providing oversight, making critical decisions, and ensuring the quality of the final product. This approach is particularly important in ontology engineering, where the nuanced understanding of domain experts is necessary to guide AI outputs and to address complex, context-specific issues.

In the MuseIT project, the OntoChat system exemplifies this human-in-the-loop approach, combining AI-driven CQ extraction with expert validation and refinement. This synergy between AI and human expertise enables the creation of more accurate and contextually relevant ontologies, such as the MultiMO, that are better suited to meet the needs of diverse user communities.

1.6 The Multimodal and Multisensory Ontology (MultiMO)

The culmination of these technological advancements and methodologies is the Multimodal and Multisensory Ontology (MultiMO), a novel ontology designed to address the specific challenges associated with annotating multisensory and multimodal datasets. MultiMO aims to provide a comprehensive framework for representing the diverse types of data encountered in cultural heritage contexts, ensuring that these rich and varied forms of knowledge are accessible and usable.

To evaluate the effectiveness of MultiMO, the MuseIT project plans to conduct two key studies: one involving the annotation of a subset of Wikidata containing multimodal cultural heritage items, and another involving an ethnographic study on how ontology engineers interact with generative AI tools. These evaluations will provide valuable insights into the practical applications and implications of the MultiMO ontology, as well as the broader impact of AI on ontology engineering practices.

In summary, the integration of generative AI techniques into ontology engineering, exemplified by LLMKE and OntoChat, represents a significant evolution in the field. Through the development and evaluation of the MultiMO ontology, the MuseIT project aims to contribute to this ongoing transformation, enhancing the ways in which knowledge is structured, accessed, and shared in the digital age.

2. Towards Automated Ontology Engineering with Human-in-the-loop

2.1. The Promise and Challenge of Large Language Models for Knowledge Engineering

To integrate generative AI successfully and effectively into the ontology lifecycle it is key not only to identify the areas where this offers most potential, but also to understand the barriers to use and form a clearer understanding of how knowledge engineering stakeholders engage with LLMs.

During August 2023 we explored a hackathon where KE workers used LLMs to solve KE tasks, including those particularly focused on multimodal areas.

The hackathon was a collaborative, interdisciplinary sprint-style research endeavor hosted by King's College London. Researchers and practitioners from several universities and both professional and academic backgrounds tackled the mission to prototype novel ideas, methods, tools, and evaluation frameworks. Hackathon participants worked in groups focused on various tasks to develop innovative integration of LLMs in the Knowledge Engineering process to not only produce and access knowledge but also to ensure its authenticity and reliability.

To understand how those involved in KG construction view the perils and promises of using LLMs, we devised a multimethod qualitative study to gather the input of the hackathon participants.

Our initial study answers the following research questions:

• How do people working in knowledge engineering perceive the promise of using LLMs to support knowledge acquisition, multimodal knowledge graphs and quality assessment?

• What do these users perceive as limitations?

Below we present the views and experiences of hackathon participants using LLMs to augment KGs. Although in theory LLMs offer potential to address some of the key challenges in KE, this is not an unmixed solution. LLMs can extract not only concepts but also relationships, constraints, parameters and more, which is very beneficial for knowledge acquisition. They also prefer facility in working with multiple types of content. Recent advances in "large action models" that can interact with web interfaces to perform online activity promise even further integration opportunities for multimodal output [1]. In particular, if LLMs can assist humans to perform evaluation faster (or in an entirely novel manner) and with more objectivity, this offers a real opportunity for improvement on the current state of play. However, there is an "abundance of caution" amongst the hackathon participants we interviewed regarding the use of LLMs in KE. This was derived not only from concern over the well-documented accuracy and trust issues inherent in LLMs, but also because they felt that some issues - such as those of evaluation - would be hard to address in an automatic fashion.

2.2 Large Language Models and Knowledge Engineering –State of the Art

KG construction has changed dramatically over the years, using semi-automated processes relying on deep-learning models and vast collections of heterogeneous data sources to scale [2]. The deep-learning models have been used to support KE for knowledge extraction, KG refinement, and KG enrichment.

- Knowledge extraction requires manual and automated approaches (supervised and unsupervised) from the respective domain. LLMs are used to identify name entities in a text, linking people,

organisations and locations [3] to associated entities in a KG with mentions of entities in text [4], and to identify the relation between entities in a text [5].

- KG refinement refers to techniques for KG completion and correction [6, 7]. For completion, the aim is to add missing knowledge to the graph. Automatic methods using LLMs fill in missing edges between KG entities and add new entities [6].
- In correction, the aim is to correct existing entities and relations. KG corrections are possible using LLMS for evaluating whether relations are true or using reasoning techniques examining the ontology axioms to identify inconsistencies [8]. KG enrichment can incorporate ontology refinement and alignment to improve a given KG. In addition, for alignment, LLMs have been used to identify and match entities between KGs, for example, using lexical matching [9].

2.3 Findings

Table 1 shows the hackathon topics with associated areas of exploration and tools utilised.

Hackathon topic	Problem area	KE Tools/Methodologies	LLM and Tools/Methodologies
Determine if LLMs can extract knowledge structures, including inference rules, to go with facts for KG construction	Knowledge acquisition, Multimodality	Investigate triples	ChatGPT, Few-shot prompt
Create a framework providing tools for collaborative human-AI ontology engineering	Knowledge acquisition, Quality assessment	Competency Questions, eXtreme Design methodology	ChatGPT, Multiple prompting techniques
Determine how KE tasks can be supported from LLMs	Knowledge acquisition, Quality assessment	NeOn methodology, Competency Questions, HermiT Reasoner, OOPS	PaLM, Llama, ChatGPT, Few-shot prompt
Determine if LLMs perform reasoning tasks completely in natural language	Knowledge acquisition, Quality assessment	Investigate triples	ChatGPT, Multiple prompting techniques
Determine if LLMs can perform ontology alignment (i.e. identify and match entities between ontologies)	Knowledge acquisition	OAEI	ChatGPT, Zero-shot prompt
Determine if multimodal LLMs can be used towards the construction of multimodal KGs	Knowledge acquisition, Multimodality	Investigate triples	mPLUG-Owl, InstructBLIP, Text only prompt (no text + image)
Determine if we can perform ontology refinement (i.e. techniques for KG completion and correction) using LLMs	Knowledge acquisition, Quality assessment, Multimodality	OntoClean	ChatGPT, Llama, Claud, Few-shot prompt

Our larger study will eventually address what the participants were able to achieve developing outputs in the hackathon using LLMs for KE. These interim results focus mainly on participants' experience of co-creating with LLMs and how that has informed their views on future use of LLMs in KE.

2.3.1 Knowledge Acquisition

There was an emphasis on the importance of datasets as a starting point. A key concern was around the trustworthiness of data provided by LLMs. One such view was that it was simple to get data just for testing purposes, but an inability to assess the accuracy of the data was a barrier to use. However, the ability to acquire data from other sources is still a challenge, so the potential of LLMs in terms of extracting, concepts, relationships, constraints or parameters or universal restrictions or existential restrictions is still an exciting possibility.

Acquiring a dataset from LLMs is highly dependant on effective prompting, or describing exactly what is required. Acquiring a dataset from LLMs is highly dependent on effective prompting or describing exactly what is required. The iterative testing of prompts is time consuming, and KE workers with certain backgrounds, for instance, natural language processing (NLP), may find this process easier than those who specialise, for instance, in semantic web.

The inability to manage LLMs to produce consistent responses, and the lack of control over the output, call for novel approaches to assessment of output by users. This was particularly problematic when the process was automated with thousands of prompting iterations.

Additionally, there was concern that syntactical errors created by the LLMs may break scripts and prevent automation when using more unstructured data. One solution to reduce syntactical errors could be to

design a robust schema with well-defined constraints for the ontology. By providing LLMs with a clear structure, such as predefined relationships and class hierarchies, the likelihood of generating syntactically incorrect scripts or data annotations would likely decrease.

2.3.2 Multimodality

Overall participants felt that LLMs did improve the potential for Multimodal Knowledge Graphs and were optimistic about the possibilities for images, sound and text integration. However, this also raised awareness of the increased human skills needed, particularly for evaluation. Currently, LLMs do not seem to offer simplification of the challenges of multimodality. One participant expressed that, all the issues experienced by other participants were simply exacerbated by those working in multimodal areas.

2.3.3 Quality Assessment

A key factor in output evaluation is the similarity of the results. Output is considered as false even if it is close to or similar to the truth. The probabilistic element of LLMs was therefore considered a barrier.

Competency questions are used in quality assessment to identify whether a KG is complete for a particular domain. More details on the LLM competency question approach can be found in Section 3. LLMs may have potential for developing a range of questions, from which users can then select the most promising. In this way, the LLM use focuses on human satisfaction with the range of questions, rather than relying on LLMs to optimise the questions. A number of respondees in our research suggested that simply attempting to replace the human process of quality assessment with LLMs was inappropriate, and new metrics may have to be developed.

While quality assessment focuses on the end output of a KG, participants also were aware of challenges with evaluating the quality of LLM-assisted individual components of knowledge graphs. Some focused on the ontology design suggesting the development of a set of ontologies to be used as gold standards. Others suggested the creation of toolkits for LLMs to review ontology errors similar to existing ontology toolkits, like OntoClean [10] and OOPS [11]. However, they acknowledged that this cannot be applied to large-scale KG like Wikidata, and manual work will still be required. Even with the capacity to automatically evaluate generative text, LLM outputs are not all the same type, for example, a KG includes dates, values, images, URLs etc. Interviewees felt it is not currently possible to have a unified way to evaluate the different types. One way to potentially alleviate this problem could be the use of multiple models e.g. validate the output from the LLMs with another model, which could act as a metadata layer to enrich the original output.

2.5. Methodology

Attendees at the hackathon were selected from applicants based on their background and experience in KE. They worked in European research labs with a strong profile in either publishing in KE venues, or in offering popular KE industry products. In total the 39 participants were from 15 different institutes and various environments from academia and industry, with 31 PhD students, 2 postdocs, 1 lecturer, 2 professors, and 3 industry members. Participants split into 7 groups of 5 to 7 members, based on their prior experience in KE. Each group investigated one of the topics presented in Table 1.

2.5.1 Data Collection

During the hackathon, an ethnographic observation study of the participants working on the challenges in groups was undertaken, using the “observer as participant” technique. After the hackathon, we

contacted 14 attendees who had volunteered to participate in semi-structured interviews. For more details on data collection, please refer to: https://kclpure.kcl.ac.uk/portal/en/publications/the-promise-and-challenge-of-large-language-models-for-knowledge-

2.5.2 Data Analysis

We conducted an initial thematic analysis across data from the interviews and ethnography. We used an inductive method to search for key themes from the interview topic guide. We then used a deductive approach to identify grounded themes that emerged in the data. For more details on data analysis and the interview topic guide, please refer to: https://kclpure.kcl.ac.uk/portal/en/publications/the-promise-and-challenge-of-large-language-models-for-knowledge-

2.4. Summary

• Knowledge Acquisition: LLMs show promise but are not trusted, and their use for knowledge acquisition is dependent on effective prompting, although this is still subject to inconsistency.
• Multimodal KGs: a barrier to use of LLMs is the need for both LLM and multimodal skills in order to use these effectively.
• Quality Assessment: LLMs can be useful for creating competency questions, and other simple tasks, but in complex scenarios, the output cannot be controlled. Therefore, humans need to stay in the loop, and also consider new forms of evaluation.

3. OntoChat: Supporting User Stories and Competency Questions with Conversation Ontology Engineering

Ontology engineering (OE) in large projects poses a number of challenges arising from the heterogeneous backgrounds of the various stakeholders, domain experts, and their complex interactions with ontology designers. This multi-party interaction often creates systematic ambiguities and biases from the elicitation of ontology requirements, which directly affect the design, evaluation and may jeopardise the target reuse. Meanwhile, current OE methodologies strongly rely on manual activities (e.g., interviews, discussion pages). After collecting evidence on the most crucial OE activities, we introduce OntoChat, a framework for conversational ontology engineering that supports requirement elicitation, analysis, and testing. By interacting with a conversational agent, users can steer the creation of user stories and the extraction of competency questions, while receiving computational support to analyse the overall requirements and test early versions of the resulting ontologies. We evaluate OntoChat by replicating the engineering of the Music Meta Ontology, and collecting preliminary metrics on the effectiveness of each component from users.

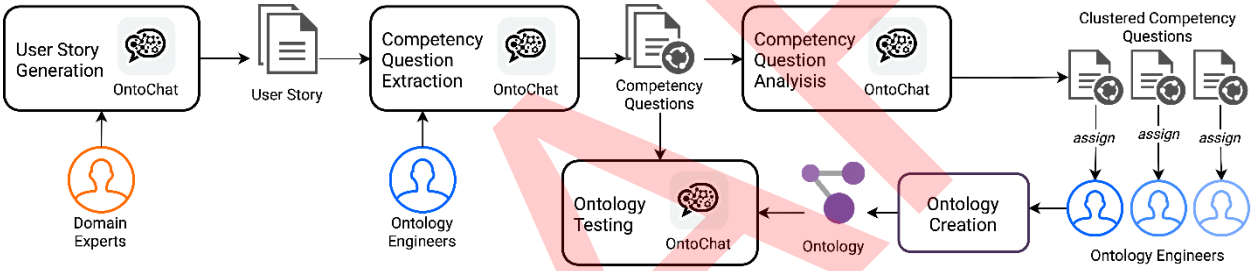

Figure 1: Illustration of the workflow alongside the main feature in OntoChat.

3.1. Function 1: Assisted persona and story creation

To enable the collaborative user story generation OntoChat follows steps below.

- Step 1: LLM role definition (back-end). The LLM behavior is primed to emulate a knowledge elicitator aiming to gather information from the user about each user story component – the persona definition, specification of concrete goals & addressed scenario, and the provision of data examples.
- Step 2: Knowledge elicitation (user involvement). The user is guided through a series of questions focusing on a particular story aspect. For instance, to gather insights about the persona's background, OntoChat asks "What are the name, occupations, skills, interests of the user?" Whenever the user does not provide enough details or gives partial answers, the LLM continues to elicit additional insights until all necessary information is gathered.
- Step 3: User story generation (back-end) The initial user story draft is created following a one-shot learning approach [14]. The LLM is supplied with a user story example and is prompted to follow the same structure for generating a user story draft summarizing the information extracted in the elicitation step.
- Step 4: Refinement (user involvement). The user is presented with the user story draft and is encouraged to provide feedback. The refinement stage is iterative and continues until the user no longer requests further changes. Possible refinements include the correction of factual inconsistencies, additions to the user stories, removal of irrelevant details, etc.

3.2. Function 2: Competency question extraction

The main objective of this module is to assist ontology engineers in extracting CQs from user stories. The procedure is organized as follows.

- Step 1: Instructing the LLM for CQ extraction (back-end). The model is provided with examples of pairs of user story fragments and expected CQs, to align its outputs to the expectations of ontology engineers.
- Step 2: First extraction of CQs (user involvement). The ontology engineer is asked to provide a user story for CQ extraction. The user story may be manually crafted or obtained from the previous step. As output, OntoChat provides an initial list of CQs.
- Step 3: Competency question refinement (back-end). The LLM is provided with a series of prompts (hidden from the user) to perform two refinement steps.
    - Step 3.1: Split not atomic CQs. If the example data is complex, users may get nonatomic questions from a single example. As these usually entail nested requirements, complex CQs need to be split. Following a few-shot learning approach, the LLM is asked whether each CQ has a complex form, hence triggering the simplification. For example, from the data "The musical work Penny Lane has genre/style baroque pop and psychedelic pop.", the LLM generated "What genres/styles are associated with Penny Lane?". After this step, the LLM replaces it with two distinct CQs: "What genres are associated with Penny Lane?" and "What styles are associated with Penny Lane?".
    - Step 3.2: Named entities abstraction. As an example, the previous CQs replaced the specific genres with the interrogative pronoun "what" (genres/styles). However, it did not remove the real-world entity "Penny Lane". Within this step, the LLMs is prompted to check again the CQs, and, guided by examples, remove possible named entities. This yields abstract CQs like "What genres are associated with the musical work?" and "What styles are associated with the musical work?".
- Step 4: User confirmation (user involvement). Finally, OntoChat asks the user whether the number of CQs and their formulation are sound. If not, by leveraging knowledge acquired from previous prompts (see Step 3), the model repeats the refinement steps until the user is satisfied.

3.3. Function 3: Competency question filtration and analysis

As some CQs may be redundant or show negligible semantic variations that are of little interest to ontology engineers, OntoChat provides support for their filtration and analysis. This is achieved through: paraphrase identification, to remove equivalent CQs; and CQ clustering, to identify groups of similar requirements. In [15], the former was found to have two benefits: (i) it mitigates the noise, and the artefacts introduced in the previous steps; (ii) it reduces the number of CQs that will be presented to ontology engineers.

In contrast to [15,14], which both rely on sentence embeddings and specialized models, this functionality is entirely supported by LLMs, which is motivated by recent findings demonstrating that LLMs possess clustering capabilities [16,17,18]. Given a list of CQs, the LLM is asked to remove redundant questions and find meaningful groups of CQs sharing the same thematic focus and intent. The latter is expected to support ontology designers in understanding requirements and possibly organizing their Agile teams (e.g.,

a team receiving a CQ cluster based on their familiarity with the sub-domain). In the current version, this step does not require user supervision.

3.4. Function 4: Ontology testing support

While the previous functionalities focus on requirement elicitation and analysis, this component provides support for testing preliminary or iterative versions of an ontology. Ontology testing efforts are often categorized into three methodologies: CQ verification, inference verification, and error provocation. The first two are concerned with verifying the correct implementation of a requirement, whereas the latter is needed to find cases where the ontology should fail [19]. These are typically done by formalizing CQs into SPARQL queries.

To test preliminary versions of an ontology, we aim for a SPARQL-free approach to achieve fast CQ verification and inference, while supporting error provocation. This is achieved in two steps: ontology verbalization, and prompt-driven CQ unit testing. Our verbalization converts an OWL ontology into plain text by documenting classes, properties, named entities, and their relationships in a description manner. The method follows a simple algorithmic procedure and assumes that the ontology is well commented to produce an expressive verbalization. Then, using the verbalization, the LLM is prompted to assess the coverage of each CQ by replying Yes/No. To prevent prompt leakage and ensure independence in the model's predictions, this is done separately for each CQ.

3.5. Evaluation

We performed a component-based evaluation of OntoChat to measure the effectiveness of each functionality and collect user feedback based on their experiences. The evaluation was organized to replicate the OE activities of the Music Meta ontology [20]. It was chosen as a benchmark/testbed for three reasons: it required considerable OE efforts and was already the source of ambiguities in the Polifonia project [21]; it was complemented by high-quality material (user stories, CQs, documentation, queries, etc.) from [13] to use as ground truth; the authors had access to a pool of domain experts for evaluation. To ensure each component is evaluated individually by the intended target users, we evaluate the more generative components by collecting feedback on their use from domain experts and ontology engineers through questionnaires.

Feedback collected from domain experts confirmed that the user stories generated with OntoChat captured the intended goal and requirements and always provided relevant information. More than 80% of participants enjoyed using the tool and found the final stories well-structured and easily understandable. While users acknowledged the model's ability to improve intermediate drafts through their feedback, only 50% were satisfied with the example data generated. Overall, 4/6 experts recognized the tool's potential to accelerate this task (2 NR), and 5 of them would prefer it over manual curation.

From the evaluation with ontology engineers, OntoChat was found to generate CQs that are comprehensive, reflective of the intended ontology scope, and easy to understand. However, 2/8 participants noted the extraction of entities outside the story's scope; and only 50% observed the potential to reduce possible author bias. While 6/8 participants expressed satisfaction with OntoChat and recognized its time-saving benefits, all agreed it holds promise for streamlining CQ generation, indicating a preference over fully manual creation.

The clustering feature proved advantageous for understanding and organizing ontology requirements when compared to full manual inspection. Participants found the interaction intuitive (62.5%) and the resulting clusters expressed meaningful groupings of CQs (87.5%). While the feature offers time-saving

benefits by providing an aggregated view of ontology requirements, there were indications that it may not fully support comprehensive analysis on its own.

For more information of OntoChat, including survey results, implementation details, evaluation results, etc., one can refer to the original paper [12]. We maintain an online space on Hugging Face for research convenience (https://huggingface.co/spaces/b289zhan/OntoChat).

3.4.1. Integrating user needs into OntoChat: A participatory prompting study

To improve the user experience and develop pre-defined prompt templates tailored to user queries for the current OntoChat user story creation workflow, we conducted a three-step participatory prompting process [22]: (1) story description: participants wrote stories based on their ontological needs; (2) prompt development: participants posed queries to request LLM support in writing user stories. Researchers refined these queries using pre-identified prompting strategies, then asked the LLM for answers, with participants evaluating the LLM-generated responses and providing feedback for further refinement; (3) story evaluation: participants assessed the stories for usefulness, clarity, and inspiration, resulting in more detailed and practical user stories.

The study involved 10 participants, including MuseIT practitioners aiming to build an ontology for multi-sensory representation in cultural heritage and PhD researchers working on knowledge engineering. They have expertise in areas including machine learning, human resources, mixed reality, implantable medical devices, multimodal representation learning, and responsible AI. Ethical clearance was obtained from the Department of Informatics at King's College London, ensuring that no personal data was collected throughout the experiment.

After coding and analysis, the collected data, we identified a list of user queries and related prompts that can be used for enhancing the effectiveness of user story generation and refinement during the human-LLM interaction. Here we provide two examples. One is on persona, for user query "detail the persona trait", the suggested prompt can be "Could you improve the persona by detailing [this specific trait] that are necessary for helping in writing an ontology user story? Explain why (this specific trait) are important." Another example is on scenario, where users have query on "narrative story for the scenario", the prompt is provided as "Could you narrate a detailed story about how the persona currently performs tasks related to their goal and how the system will enhance this process? Please include practical insights and a step-by-step sequence of actions.". We documented the complete list also in OntoChat repository (https://github.com/King-s-Knowledge-Graph-Lab/OntoChat), with a detailed explanation available in the original paper [33]. The identified prompts are integrated into OntoChat for future work.

To complement our results for ontology testing, we report the confusion matrix in Fig. 6, expressing the number of correct predictions (25 true positives, 24 true negatives) and wrongly classified competency questions (3 false positives, 4 false negatives). Please, note that this can be seen as an instance of competency question verification and error provocation for positive and negative CQs, respectively.

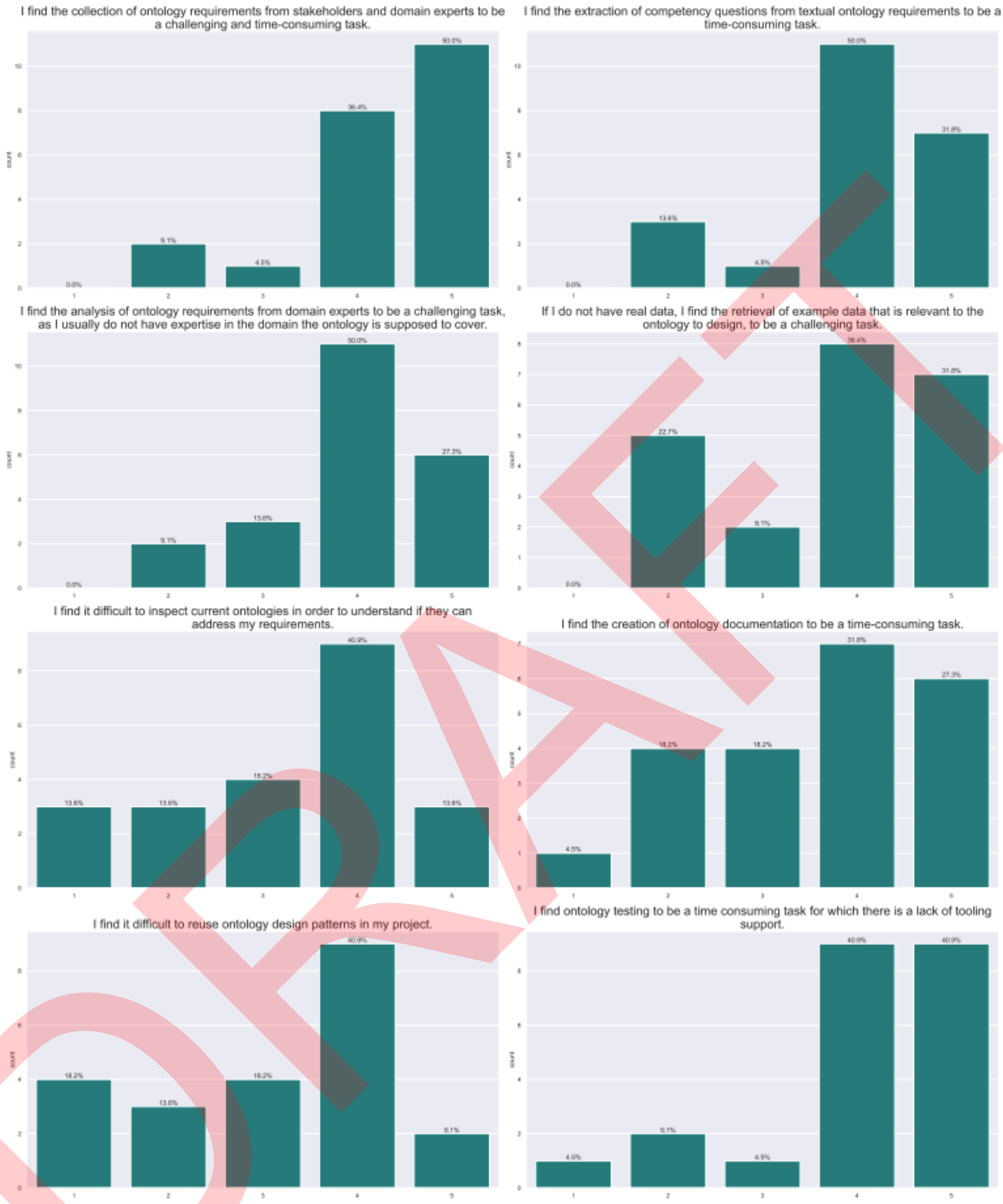

Figure 2: Responses from the Ontology Engineering survey. Replies quantify the agreement of participants with respect to each statement on a 5-point Likert scale, where 1 (absolutely disagree) to 5 (absolutely agree), with 3 being a neutral response (neither agree nor disagree).

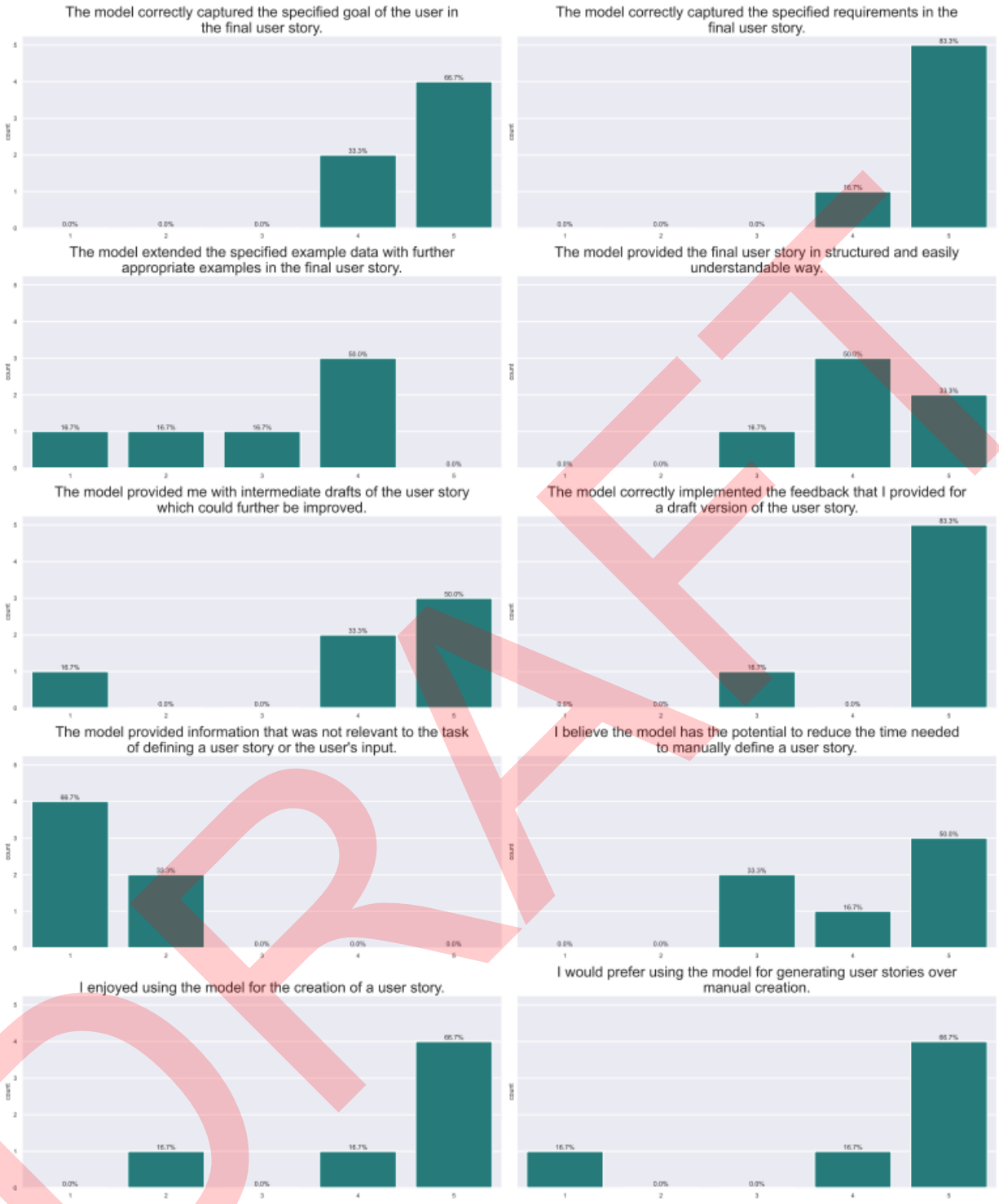

Figure 3: User ratings on the *Collaborative User Story Creation* functionality of OntoChat. The evaluation was performed by N=6 domain experts.

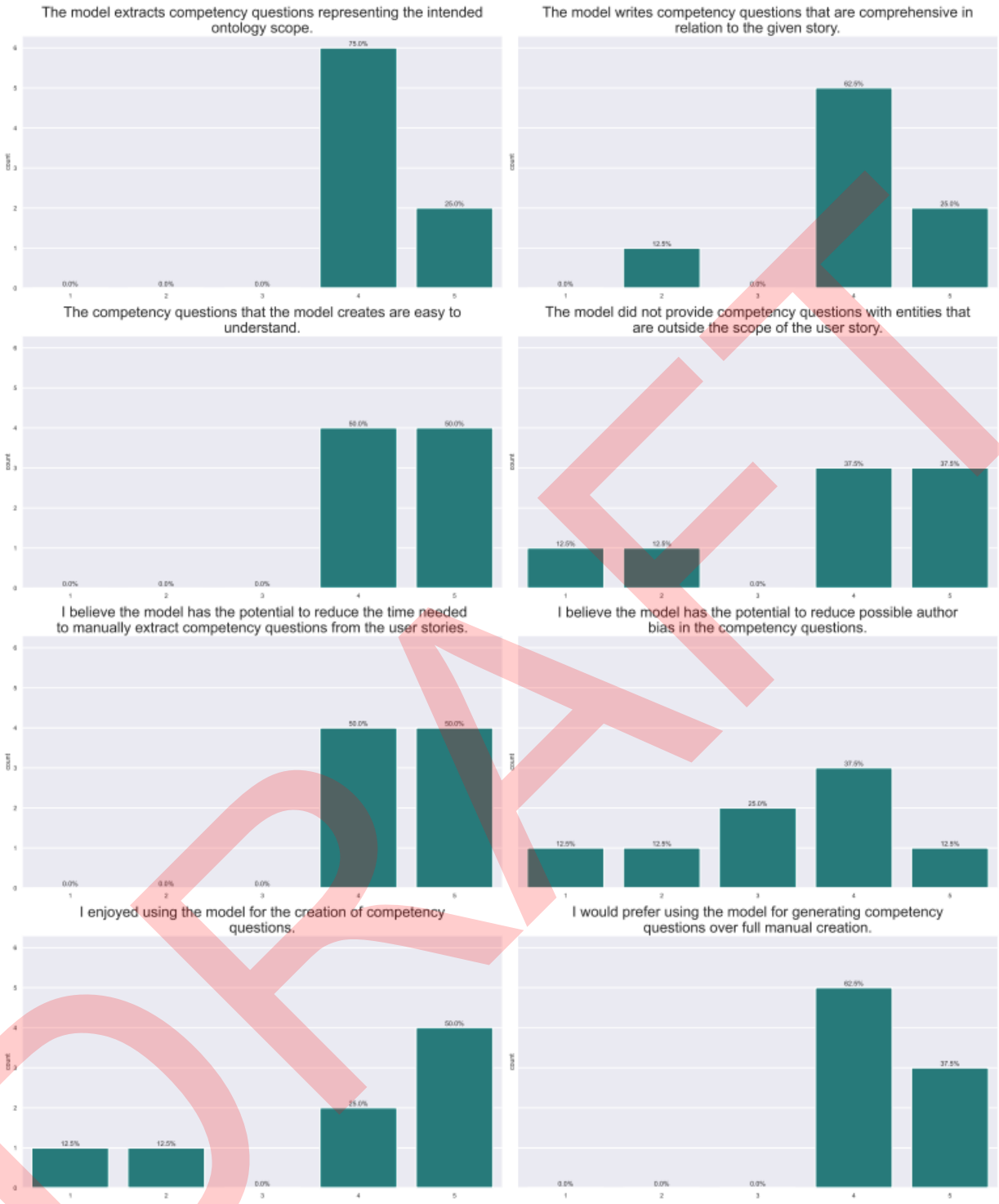

Figure 4: User ratings on the *Competency Question Extraction* functionality of OntoChat. This evaluation was performed by N=8 ontology engineers.

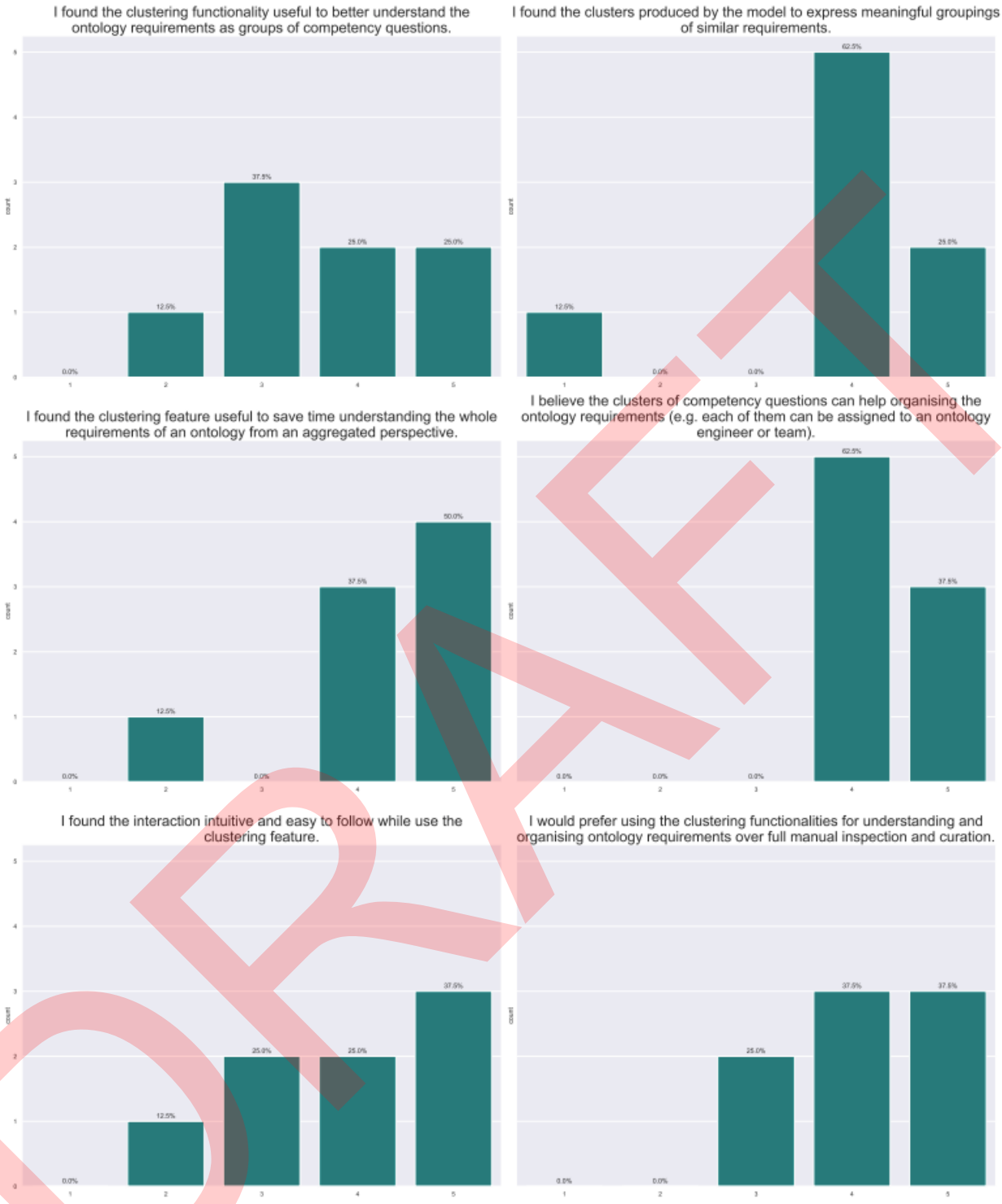

Figure 5: User ratings on the *Competency Question Clustering* functionality performed by N=8 ontology engineers.

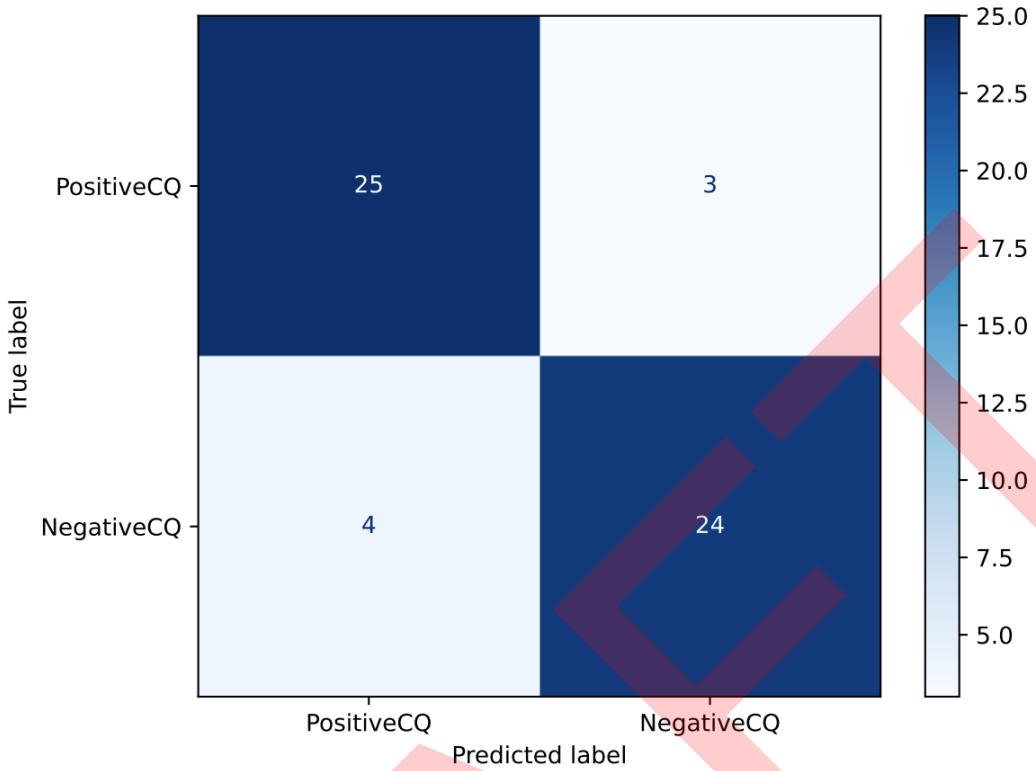

Figure 6: Confusion matrix summarising our results for *prompt-driven CQ unit testing*. Positive CQs (target label 1) denote competency questions that are expected to be addressed or covered by the ontology, whereas negative CQs (target label 0) are artificially created to express requirements that are not yet addressed by the ontology and should thus be predicted as such.

3.6 Conclusion

This work addresses the challenges of ontology engineering in large collaborative projects by implementing a conversational workflow to streamline the process. The proposed framework, OntoChat, leverages LLMs to facilitate requirement elicitation, analysis, and ontology testing. Our preliminary evaluation efforts demonstrate a positive response from domain experts and ontology engineers, indicating potential for accelerating conventional ontology engineering tasks. Nonetheless, several limitations still exist, notably those inherent to the use of LLMs in specialised domains due to their limited or potentially obsolete knowledge. Additional challenges include addressing biases in persona creation and enhancing the framework to provide insights into implementation costs and timelines. This will allow us to measure the amount of user supervision and involvement (e.g., number of interactions with the LLM, specificity of user feedback) during the refinement steps, needed to achieve a reasonable output from OntoChat (e.g., a user story, a list of competency questions), in contrast to full manual curation. Future work will focus on addressing these challenges, while enhancing the generation of examples in user stories, refining named entity scope in competency question creation, and broadening analysis support.

4. LLMKE: Using Large Language Models for Knowledge Engineering

4.1. Introduction

Language models have been shown to be successful for a number of Natural Language Processing (NLP) tasks, such as text classification, sentiment analysis, named entity recognition, and entailment. The performance of language models has seen a remarkable improvement since the advent of ChatGPT [23] and GPT-4 model [24], which induced the development of several other LLMs such as LLama from Meta [25], Claude from Anthropic, and Bard from Alphabet.

This surge in the development and release of large LMs, many of which have been trained with Reinforcement Learning with Human Feedback (RLHF) [26], has allowed users to consider the LMs as *knowledge repositories*, where they can interact with the models in the form of `chat' or natural language inputs. This form of interaction, combined with the unprecedented performance of these models across NLP tasks, has shifted the focus to the engineering of the input, or the `prompt' to the model in order to elicit the correct answer. Subsequently, there has been a steady increase in research outputs focusing on prompt engineering in the recent past [27, 28].

The idea of using LLMs to construct and complete knowledge graphs (KGs) has been investigated by many studies [26, 29, 30]. However, the recent boost in performance has once again brought to the surface the question of using LLMs for, or even as, KGs.

Despite the incredible promise of LLMs as knowledge stores, there are fundamental differences that set them apart and even disadvantage them as compared to KGs. The reasoning and inference power of the KGs are the most important of these differences. Not only do traditional KGs store facts, they also impose logical constraints on the entities and relations in terms of defining the types of the entities as well as prescribing the domain and range of the relations. Additionally, the most popular and successful LLMs have been trained on data obtained from publicly available sources, and due to the inherent limitations of the training method of these models, they tend to exhibit expert-level knowledge in popular domains while being largely ignorant of the lesser-known ones.

In this section, we describe our approach to using LLMs for Knowledge Engineering (KE) tasks, especially targeting solving the ISWC 2023 LM-KBC Challenge, and report our findings regarding the prospect of using these models to automate the process of KE. The task set by this challenge is to predict the object entities (zero or more) given the subject entity and the relation that is sourced from Wikidata. For instance, given the subject *Robert Bosch LLC* with Wikidata QID *Q28973218* and the property *CompanyHasParentOrganisation*, the task is to predict the list of object(s) *Robert Bosch* (*Q234021*). We used two state-of-the-art LLMs, gpt-3.5-turbo and GPT-4 [2] for this task. By performing different experiments using few-shot approaches, as well as leveraging appropriate context, we have been able to achieve a macro-average F1 score of 0.689 (0.7007 on CodaLab), with F1-scores ranging from 0.3282 in the *PersonHasEmployer* property to 1.0 in the *PersonHasNobelPrize* property.

4.2. Methods

4.2.1 Problem Formulation

Most of the previous works [26, 29, 30] on using LLMs for fact completion stop at the string level, which leaves gaps for constructing hands-on knowledge graphs and thus hinders downstream application. Our work pushed a step forward on this task, where the extracted knowledge is not only in string format but also linked to their respective Wikidata entities. Formally, given a query consisting of subject entity s

and relation r, the task is to predict a set of objects $\{o_i\}$ with unknown numbers ($|\{o_i\}| \geq 0$) by prompting LLMs and mapping the objects to their related Wikidata entities $\{w_{o_i}, \cdots, w_{o_n}\}$.

4.2.2 LLM-based Knowledge Engineering (LLMKE) Pipeline

Knowledge Probing

The pipeline consists of two steps: *knowledge probing* and *Wikidata entity mapping*. For the knowledge probing step, we engineered prompt templates for probing knowledge from LLMs. We adopt OpenAI's gpt-3.5-turbo and GPT-4 [2] in this step. For each of the LLMs, we run experiments with three types of settings. The first is question prompting, where LLMs are provided with questions as queries. For example, ''Which countries share borders with Brazil?''. The second is triple completion prompting, where prompts are formatted as incomplete triples, such as ''*River Thames*, *RiverBasinsCountry*:''. There are several heuristics employed in these two settings. For example, there are only 5 different Nobel Prizes, so *PersonHasNobelPrize* has 6 candidate answers, including the empty answer. Providing all potential answers at the prompt is likely to help LLMs perform well (F1-score close to 1) and return the desired format.

In the third setting, we provide context to help LLMs by enriching knowledge. In the first step, we ask LLMs to predict the objects based on their own knowledge using the same settings as question prompting. Then we provided the context, and we let LLMs predict again by considering the context and comparing it with their own predictions. In this study, we used Wikipedia as the general context corpus. The first paragraphs of the entity's Wikipedia page (the introduction) and the JSON format of the Wikipedia Infobox are organized and provided to LLMs. LLMs were asked to make predictions again by considering the context and comparing it with the previous response.

In all settings, we perform few-shot learning, where we first provide three examples. Since the required format of results is a list, providing examples with the exact format is expected to help LLMs return better-formatted results. In the dataset, there are some relations that could potentially have empty results. In this case, the prompt indicated the required return format (i.e., [‘’’’]).

Wikidata Entity Mapping

The entity mapping step first finds Wikidata entities for each object string using the MediaWiki Action API. One of the actions, *wbsearchentities* which searches for entities using labels and aliases, returns all possible Wikidata entities as candidates. Then, in the disambiguation step, the actual Wikidata entities linked to the objects are selected. To reduce the cost while improving the accuracy for disambiguation, we treated different relations with three methods: *case-based*, *keyword-based*, and *LM-based*.

The *case-based* method is a hard-coding solution for efficiently solving ambiguities for relations with smaller answer spaces and limited corner cases. For example, *CompoundHasParts* only has all the chemical elements as its answer space. Further, it only has one ambiguous case: ‘*mercury*’. The case-based method always maps ‘mercury’ in the object lists to Q925 (the chemical element with symbol Hg) instead of Q308 (the planet). For other relations with a larger answer space but also entities with common characteristics, we used the *keyword-based* method, which extracts the description of the entity and searches entities with their description using relevant keywords. This method is used when there are common words in the entity description. For example, object entities of the relation *CountryHasOfficialLanguage* always have the keyword ‘language’ in their descriptions.

The above two methods clearly suffer from limitations due to their poor coverage and inflexibility. The third method is language model-based (*LM-based*). We constructed a dictionary of all candidate QIDs with their labels and descriptions, concatenated it with the query in this first step, and asked LMs to determine which one should be selected. This method is used when there is no semantic commonality between the answers and disambiguation is required to understand the difference between entities, e.g., properties with the whole range of human beings as potential answers such as `*PersonHasSpouse*'. As there is no commonality among the labels and descriptions of answers, the decision is left to the LMs. This method also has limitations, such as being time-consuming and unstable in terms of the responses from the LLM.

4.3. Results

4.3.1 Datasets

The dataset used in the ISWC 2023 LM-KBC Challenge is queried from Wikidata and further processed. It comprises 21 Wikidata relation types that cover 7 domains, including music, television series, sports, geography, chemistry, business, administrative divisions, and public figure information. It has 1,940 statements for each train, validation, and test sets. The results reported are based on the test set. To investigate the actual knowledge gap between LLMs and Wikidata, we created ground truths of the test set for offline evaluation manually from Wikidata. The online evaluation results from CodaLab are reported in [32]. In the dataset, the minimum and maximum number of object-entities for each relation is different, ranging from 0 to 20. The minimum number of 0 means the subject-entities for some relations can have zero valid object-entities.

4.3.2 Model Performance

In terms of the overall performance of the model, GPT-4 is better than gpt-3.5-turbo. The in-context learning setting has the best performance compared with the other two few-shot learning settings. For few-shot learning, the performance on question answering prompts and triple completion prompts is quite close.

Table 2 - Comparison of the performance of gpt-3.5-turbo and GPT-4 models.

Model	question			triple			context		
	P	R	F1	P	R	F1	P	R	F1
gpt-3.5-turbo	0.581	0.597	0.563	0.576	0.609	0.554	0.625	0.684	0.618
GPT-4	0.682	0.689	0.661	0.678	0.683	0.657	0.676	0.709	0.665

Table 3 - The results of probing GPT-4 with few-shot examples. The 'context' represents question prompts with Wikipedia context. All results have been disambiguated. For each relation, the best F1-scores among the three settings are highlighted in bold.

Relation	question			triple			context		
	P	R	F1	P	R	F1	P	R	F1
BandHasMember	0.576	0.632	0.573	0.591	0.627	**0.581**	0.510	0.627	0.527
CityLocatedAtRiver	0.780	0.562	0.615	0.775	0.578	**0.629**	0.648	0.504	0.533

CompanyHasParentOrganisation	0.590	0.755	**0.590**	0.560	0.745	0.563	0.512	0.810	0.520
CompoundHasParts	0.782	0.976	0.837	0.782	0.964	0.835	0.787	0.981	**0.843**
CountryBordersCountry	0.802	0.685	0.730	0.806	0.688	0.734	0.829	0.723	**0.763**
CountryHasOfficialLanguage	0.956	0.854	0.883	0.949	0.858	0.883	0.938	0.873	**0.886**
CountryHasStates	0.796	0.809	0.800	0.754	0.748	0.750	0.805	0.816	**0.807**
FootballerPlaysPosition	0.685	0.693	0.680	0.710	0.733	**0.708**	0.545	0.565	0.550
PersonCauseOfDeath	0.765	0.783	0.762	0.795	0.803	0.793	0.800	0.803	**0.798**
PersonHasAutobiography	0.478	0.471	**0.461**	0.458	0.486	**0.461**	0.475	0.471	0.459
PersonHasEmployer	0.362	0.343	0.327	0.353	0.357	**0.328**	0.325	0.397	0.321
PersonHasNobelPrize	1.000	1.000	**1.000**	1.000	1.000	**1.000**	1.000	1.000	**1.000**
PersonHasNumberOfChildren	0.550	0.550	0.550	0.520	0.520	0.520	0.690	0.690	**0.690**
PersonHasPlaceOfDeath	0.670	0.730	0.670	0.730	0.730	0.690	0.783	0.810	**0.785**
PersonHasProfession	0.494	0.420	0.427	0.422	0.422	**0.444**	0.390	0.408	0.363
PersonHasSpouse	0.687	0.690	0.685	0.660	0.660	0.651	0.718	0.750	**0.727**
PersonPlaysInstrument	0.566	0.565	0.531	0.519	0.519	0.507	0.559	0.597	**0.534**
PersonSpeaksLanguage	0.747	0.813	0.744	0.836	0.836	0.759	0.757	0.808	0.742
RiverBasinsCountry	0.841	0.946	**0.855**	0.931	0.931	0.852	0.827	0.941	0.852
SeriesHasNumberOfEpisodes	0.590	0.590	0.590	0.530	0.530	0.530	0.690	0.690	**0.690**
StateBordersState	0.608	0.600	0.567	0.608	0.608	**0.581**	0.612	0.618	0.578
Average	0.682	0.689	0.661	0.678	0.683	0.657	0.676	0.709	0.665

From the lens of relations, LLMs perform well when the relation has a limited domain and/or range, for example, *PersonHasNobelPrize*, *CountryHasOfficialLanguage*, and *CompoundHasParts*. On the other hand, LLMs perform poorly for relations such as *PersonHasEmployer*, *PersonHasProfession*, and *PersonHasAutobiography*. This may be due to two reasons: firstly, LLMs have limited knowledge about public figures and their personal information (except for famous ones). Secondly, the unlimited answer space for such relations could increase the difficulty of prediction. The results show that LLMs perform well on the knowledge of geography (*CityLocatedAtRiver*, *CountryBordersCountry*, *CountryHasStates*, *RiverBasinsCountry*, *StateBordersState*), and the performance is inversely correlated with the size of the object range.

4.3.3 In-context learning

Providing context to LLMs is an established method for improving model performance [31]. As such, we experimented with various sources and forms of context and selected the best one for each relation. In particular, we experimented with using the introduction content of the Wikipedia article for the subject entity, the Infobox of the Wikipedia article for the subject entity in JSON format, as well as relation-specific sources of information such as IMDb. The effect of providing context varies for different models. It is

observed gpt-3.5-turbo benefits from the context more compared with GPT-4. In contrast to our intuition, adding context did not improve the performance of GPT-4 in all relations as compared to the few-shot setting. For most properties, where context improved the performance, the introduction and Infobox of the Wikipedia page are sufficient. Notable exceptions to the above are the *SeriesHasNumberOfEpisodes* and the *CountryHasState* relations. For the *SeriesHasNumberOfEpisodes* relation, we augmented the Wikipedia-based context with context provided from IMDb. The information on IMDb was added to the prompt prefaced by the label ‘‘IMDb’’, and the model was asked to use this information (if it was available) to provide an answer. Moreover, for the *CountryHasState* relation, we discovered that GPT-4 would treat ‘state’ more like the definition of ‘country’ than that of the administrative division entity. Therefore, we experimented with different contexts and realized that the model provided the most accurate results when provided with the Wikipedia page for the term ‘‘Administrative Division''. More information on the experimental setting for each relation can be seen in Table 2.

Table 4 - The context types and disambiguation methods used for each relation.

Relation	Context type	Disambiguation method
BandHasMember	Wikipedia Intro + Infobox	Keyword-based
CityLocatedAtRiver	Wikipedia Intro + Infobox	LM-based
CompanyHasParentOrganisation	Wikipedia Intro + Infobox	-
CompoundHasParts	Wikipedia Intro + Infobox	Case-based
CountryBordersCountry	Wikipedia Intro + Infobox	-
CountryHasOfficialLanguage	Wikipedia Intro + Infobox	Keyword-based
CountryHasStates	Wikipedia Page	LM-based
FootballerPlaysPosition	Wikipedia Intro + Infobox	Case-based
PersonCauseOfDeath	Wikipedia Intro + Infobox	-
PersonHasAutobiography	Wikipedia Intro + Infobox	Keyword-based
PersonHasEmployer	Wikipedia Intro + Infobox	Case-based
PersonHasNobelPrize	Wikipedia Intro + Infobox	-
PersonHasNumberOfChildren	Wikipedia Intro + Infobox	-
PersonHasPlaceOfDeath	Wikipedia Intro + Infobox	-
PersonHasProfession	Wikipedia Intro + Infobox	Case-based
PersonHasSpouse	Wikipedia Intro + Infobox	LM-based
PersonPlaysInstrument	Wikipedia Intro + Infobox	Case-based
PersonSpeaksLanguage	Wikipedia Intro + Infobox	-
RiverBasinsCountry	Wikipedia Intro + Infobox	Case-based
SeriesHasNumberOfEpisodes	IMDb + Wikipedia Intro + Infobox	-
StateBordersState	Wikipedia Intro + Infobox	LM-based

4.3.4 Disambiguation

When employing the baseline disambiguation method provided by the challenge, we noticed ambiguities for 13 relations in total, with the model predicting the correct string but the returned QID being different from the ground truth. To remedy this issue, we employed different disambiguation methods with increasing computational costs. Specifically, we experimented with baseline Wikidata-based, keyword-based, case-based, and LM-based disambiguation methods. The best-performing disambiguation method for each relation is shown in Table 3. From Table 4, we can observe that F1-score increases for all settings and models.

Table 5 - The results of disambiguation methods.

Model	Setting	Baseline			Disambiguation		
		P	R	F1	P	R	F1
gpt-3.5-turbo	question	0.557	0.574	0.540	0.581	0.597	0.563
	triple	0.545	0.579	0.525	0.576	0.609	0.554
	question (context)	0.599	0.659	0.593	0.625	0.684	0.618
GPT-4	question	0.650	0.661	0.632	0.682	0.689	0.661
	triple	0.641	0.651	0.624	0.678	0.683	0.657
	question (context)	0.650	0.685	0.641	0.676	0.709	0.665

4.4. Conclusion

Within the scope of the ISWC 2023 LM-KBC challenge, this work aimed at developing a method to probe LLMs for predicting the objects of Wikidata triples given the subject and property. Our best-performing method achieved state-of-the-art results with a macro-averaged F1-score of 0.689 (0.7007 online evaluation) across all properties, with GPT-4 having the best performance on the *PersonHasNobelPrize* relation and achieving a score of 1.0, while only achieving a score of 0.328 on the *PersonHasEmployer* relation. These results show that LLMs can be effectively used to complete knowledge bases when used in the appropriate context. At the same time, it is important to note that, largely due to the gaps in their knowledge, fully automatic knowledge engineering using LLMs is not currently possible for all domains, and a human-in-the-loop is still required to ensure the accuracy of the information.

5. MultiMO: An Ontology for Documenting Multimodal and Multisensory Knowledge Graphs

5.1. Introduction

The age of generative AI and the capacity of its models, such as StableDiffusion and MusicLM, to generate human-like images and music, has brought the attention of research communities and the wider public into multimodality. Multimodality is the "application of multiple literacies or 'modes' within one medium that contribute to an audience understanding of a composition".[1] In generative AI, multimodality is understood as the capacity of its models to not just generate text (e.g. as in the original ChatGPT), but also to use text ("prompts") to generate images, music, videos, and many others. This has given rise to large collections of effectively multimodal datasets, combining for example the text of prompts and its correspondingly generated images [34].

Simultaneously, this multimodal trend has been on the rise in more traditional, symbolic forms of knowledge representation, such as knowledge graphs (KGs) [38]. KGs have traditionally contained unimodal representations in the form of either URIs (global, unique, de-referenceable IDs of resources) or so-called "literals" (strings of text, numbers, dates, etc.). However, more recently collaborative approaches of KG construction, such as Wikidata [37], have supported the creation of KGs with more rich literals or media objects, like images, sounds, music, speech recordings of Wikipedia articles, or videos. This has happened because Wikidata, an open KG that everyone can edit, was built primarily as a central data-hub to support the multilingual content of Wikipedia. Starting with the structured content of infoboxes, the increasing size and coverage of Wikidata has effectively transformed it into the Wikipedia data backbone, therefore requiring the inclusion of not just dates, population numbers or countries, but also pictures, 3D models, audio recordings, and a plethora of other language-independent content.[2] The availability of these multimodal knowledge graphs poses a great opportunity for universal access to knowledge. The World Bank reports that "one billion people, or 15% of the world's population, experience some form of disability". This, in combination with the fact that the perceptual modalities of sight and hearing have been at the core, and often taken for granted, of representing digital assets, sets an urgent agenda for leveraging multimodal and multisensory content as gateways to truly universal access to knowledge for all.

However, this early availability of multimodal AI and multimodal knowledge graphs also raises many questions and challenges regarding their completeness, trustworthiness, and preservation. If the availability of appropriate modalities of knowledge for all depends on volunteer contributions, some items/modalities will likely receive more attention than others, creating biases and incompleteness problems. Generative AI can be used to complete missing information, but at the cost of concerns regarding the provenance of training data. All these issues have, at their root, the problem of lacking sufficient documentation: in general, multimodal knowledge graphs are not appropriately documented on the modalities they contain, and how they fit the access requirements of users. The long-term preservation of multimodal knowledge graphs exacerbates this problem. Existing approaches address the broader problem of providing vocabularies and metadata to document the contents of knowledge graphs, e.g. DCAT [39] (for general datasets), VoID [40] for RDF knowledge graphs, and Croissant [36] (for machine

[1] https://en.wikipedia.org/wiki/Multimodality
[2] https://www.wikidata.org/wiki/Category:Properties_with_commonsMedia-datatype

learning datasets). However, none of these combines the specification of multimodal content with access requirements, multisensory input, and disability.

Here, we propose MultiMO, a vocabulary for documenting sensory modalities in knowledge graphs. MultiMO builds on top of existing vocabularies for documenting knowledge graph contents and extends them to bring together specifications of multimodality (i.e. image, audio, video content) with sensory requirements (i.e. items that can only be consumed by users with e.g. at least 50% hearing). It does so by wrapping specific subsets of existing knowledge graphs that adhere to these multimodal and multisensory constraints. Our research questions are:

- RQ1. How can current data models of knowledge graph documentation and metadata be extended to specify multimodal and multisensory content?
- RQ2. How can such models be deployed and evaluated in current generative AI and human-in-the-loop approaches?

5.2. Related Work

Various existing works address multimodal and multisensory representations, especially in the cultural heritage domain. In [34], authors propose CUBE, a benchmark to evaluate the cultural competence of text-to-image models and address issues of cultural awareness and diversity. Bias in cultural heritage is a well-known and studied issue. In [35], authors find omissions and biases in museum collections by mapping contents of datasets to an ontology based on Wikidata and by analysing the geographical distribution of records.

Various ontologies and vocabularies have been proposed to describe the multimodal contents of datasets. In the Semantic Web, the Data Catalog Vocabulary (DCAT) is “designed to facilitate interoperability between data catalogs published on the Web” [39]. The Dublin Core Metadata Initiative (DCMI) [41] contains “fifteen terms and several dozen properties, class, datatypes and vocabulary encoding schemes” in RDF. More specifically for RDF datasets, the Vocabulary of Interlinked Datasets (VoID) [40] is “concerned with metadata about RDF datasets", providing "terms and patterns for describing RDF datasets, and is intended as a bridge between the publishers and users of RDF data”. VoID can be used not just to broadly describe the contents of RDF knowledge graphs, but also to specify concrete subsets, provide statistics, and point at specific example resources. Unfortunately, no available vocabulary incorporates the ability to specify concrete modalities supported by either datasets or partitions/subsets, nor the capacity to indicate whether such modalities can be offered to users with consumption restrictions due to disability.

5.3. MultiMO: Documenting Sensory Modalities in Knowledge Graphs

MutiMO is an ontology and vocabulary concerned with describing metadata about the multimodal and multisensory content of Knowledge Graphs. MultiMO builds on top of existing vocabularies that provide resource to describe knowledge graph metadata in general, such as DCAT and VoID; and extends them in order to provide additional multimodal and multisensory features.

Figure 7 shows a diagram of the main classes and properties of MultiMO. Beyond just adding the capacity to describe the multimedia types supported in the entities of a knowledge graph, which is typically

achieved via simple DCAT properties such as *dcat:mediaType*, MultiMO provides an abstraction that encapsulates both **multimodality** and **multisensory** information at the same time. This is achieved via the *mmo:SensoryModality* class, a specialisation of the *void:Dataset* class and particularly a *void:subset*. The idea with this design is threefold:

1. Explicitly model **modalities in combination with a sensory specification**.
2. Allow the multimodal and multisensory specification of a **subset or part** of a knowledge graph
3. Keep backwards **compatibility** with other knowledge graphs described with standards such as DCAT and VoID.

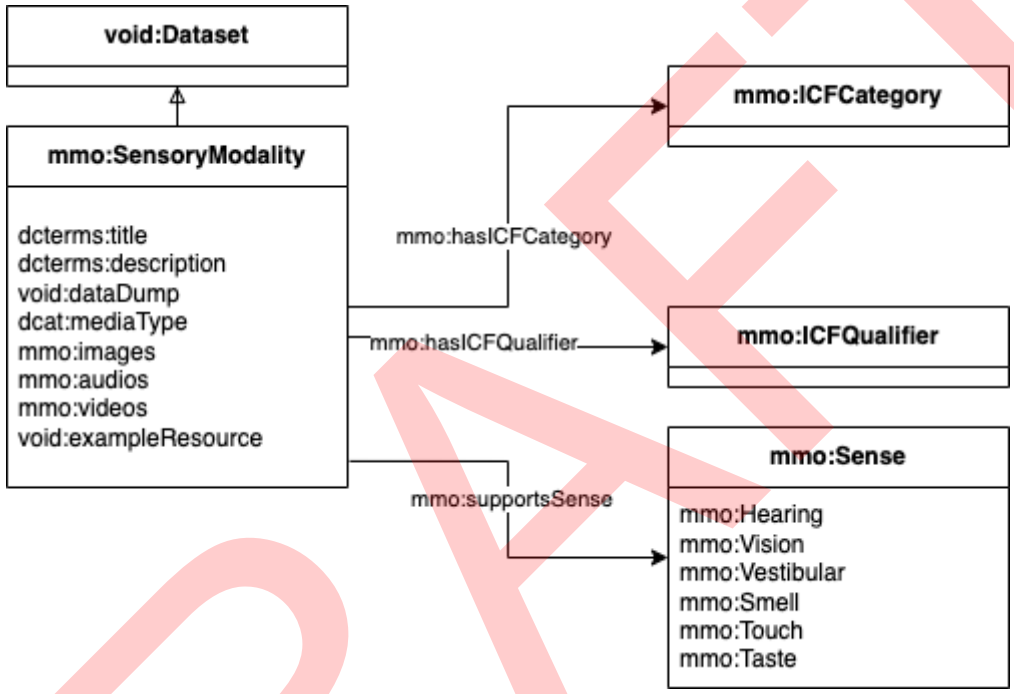

Figure 7: The MultiMO data model for multimodal and multisensory knowledge graphs.

Therefore, the *mmo:SensoryModality* class provides properties to describe a knowledge graph partition that contains resources of certain media types (images, sounds, videos, 3D models, haptic maps/feedback, etc.) that are compatible with the sensory capabilities of users as described by the following set of properties:

- Senses: this is the classic taxonomy of senses as provided by Wikipedia,[3] and containing 6 instances for Hearing, Vision, Vestibular (i.e. equilibrioception as perceived by the physical stimuli of gravity and acceleration), Smell, Touch and Taste. A *mmo:SensoryModality* specification can indicate any number of senses supported in it through various uses of the *mmo:supportsSense* property and instances of this taxonomy.

[3] https://en.wikipedia.org/wiki/Sense

- ICF Category and ICF Qualifier: MultiMO explicitly supports WHO's ICF, the International Classification of Functioning, Disability and Health (ICF),[4] by allowing users to indicate an *mmo:ICFCategory* (via the *mmo:hasICFCategory* property) and qualify such category with a *mmo:ICFQualifier* (via the *mmo:hasICFQualifier* property). The idea behind this is that a *mmo:SensoryModality* is characterised by one body function, mental function, voice/speech function, neuromusculoskeletal function, etc. (category) being affected or constrained by a barrier or facilitator, capacity, magnitude of impairment, etc. (qualifier). ICF categories contain a broad spectrum of body and mental structures commonly discussed in disability, and up to 13 different qualifiers (containing e.g. no barrier, mild, moderate, substantial, severe, complete barriers, with different degrees of quantified limitation (from 0% to 100%).

On top of this sensory capability specification, MultiMO subsets contain an unlimited number of statements with supported multimedia types via the *dcat:mediaType* property. In DCAT, this property can be used to specify the media type (i.e. file type) that is used to digitally represent a dataset, using the taxonomy of Media Types by the Internet Assigned Number Authority (IANA).[5] Here, we extend the use of this property to indicate the media types that are contained within the knowledge graph partition specified by the sensory information provided by *mmo:supportsSense*, *mmo:hasICFCategory* and *mmo:hasICFQualifier*.

Finally, *mmo:SensoryModality* contains a number of additional metadata properties that are inherited from VoID and DCAT:

- *dcterms:title* provides a title for the knowledge graph partition specified by this sensory modality
- *dcterms:description* is used to add a textual, human-readable description of the sensory modality partition
- *void:DataDump* is a link to access a downloadable dump of the sensory modality
- *mmo:images* is used to indicate the total number of images in the sensory modality partition. These are typically binary literals in the object position of RDF triples. This and the following two properties can also be used outside of a partition to indicate the overall number of resources of this media type in the knowledge graph
- *mmo:videos* works analogous to *mmo:images* for videos
- *mmo:audios* works analogously to the previous for audio recordings
- *void:exampleResource* is used to provide the URI of a resource that is representative and exemplifies other resources in the sensory modality partition. For example, if the partition relates to content that can only be heard, it will provide the URI of a resource that contains audio files, speech description recordings, etc.

The example below shows how to use MultiMO to specify two sensory modality partitions of Wikidata containing images and audiovisual material that can be consumed by those with mild light sensitivity:

[4] https://icd.who.int/dev11/l-icf/en#/

[5] http://www.iana.org/assignments/media-types/media-types.xhtml

```
<https://www.wikidata.org/wiki/Wikidata:Main_Page> a void:Dataset ;
        void:subset :Wikidata_images;
        void:subset :Wikidata_audiovisual .

:Wikidata_images a void:Dataset, mmo:SensoryModality;
        dcterms:title “Wikidata image modality partition for mild light sensitivity” ;
        dcterms:description “A subset of all Wikidata items that contain images as their main
        modality that can be consumed by those with mild light sensitivity” ;
        dcat:mediaType <http://www.iana.org/assignments/media-types/image/png>  , <
        http://www.iana.org/assignments/media-types/image/jpeg> ;
        mmo:images 50000 ;
        void:exampleResource <https://www.wikidata.org/wiki/Q233> , <
        https://www.wikidata.org/wiki/Q586> ;
        mmo:sense mmo:Vision ;
        mmo:ICTCategory <https://icd.who.int/dev11/l-
        icf/en#/http%3a%2f%2fid.who.int%2ficd%2fentity%2f1171742188> ;
        mmo:ICTQualifier <https://icd.who.int/dev11/l-
        icf/en#/http%3a%2f%2fid.who.int%2ficd%2fentity%2f701786589> .

:Wikidata_audiovisual a void:Dataset, mmo:SensoryModality;
        dcterms:title “Wikidata audiovisual modality partition” ;
        dcterms:description “A subset of all Wikidata items that contain audiovisual materials as their
        main modality” ;
        dcat:mediaType <http://www.iana.org/assignments/media-types/video/ogg> ;
        dcat:mediaType <http://www.iana.org/assignments/media-types/video/mpeg> ;
        mmo:videos 14402 ;
        void:exampleResource <https://www.wikidata.org/wiki/Q76436> ,
        <http://www.wikidata.org/entity/Q79833> ;
        mmo:sense mmo:Vision, mmo:Hearing ;
        mmo:ICTCategory <https://icd.who.int/dev11/l-
        icf/en#/http%3a%2f%2fid.who.int%2ficd%2fentity%2f1171742188> ;
        mmo:ICTQualifier <https://icd.who.int/dev11/l-
        icf/en#/http%3a%2f%2fid.who.int%2ficd%2fentity%2f701786589> .
```

Mappings to Wikidata properties

Various of the MultiMO properties have similar ones in Wikidata that can be mapped for enhanced compatibility. We study all Wikidata properties that take values in the domain of “commonsMedia” datatype,[6] essentially meaning multimedia content that cannot be encoded as simple text. We issue the below mappings through *rdfs:subPropertyOf* triples to the following Wikidata properties:

[6] https://www.wikidata.org/wiki/Category:Properties_with_commonsMedia-datatype

Multimodal properties	
P10	video
P18	Image (with subproperties e.g. P41 flag image)
P51	Audio (with subproperties e.g. P989 spoken text audio)
P2919	Label in sign language
P3030	Sheet music
P4896	3D model
Multisensory properties	
P5872	Smells of
No defined properties for taste	N/A
No defined properties for touch	N/A

Noticeably, Wikidata has no defined properties to describe the taste or the touch of its items, which is a limitation for users that use those senses—possibly in combination with others. Labels in sign language, sheet music, and 3D models are especially interesting for access for all as e.g. people with impaired hearing may be able to read sheet music, and people who rely on their touch can potentially use 3D models by touching physical representations of them or through haptic devices.

5.4. Preliminary Evaluation

The preliminary evaluation of MultiMO consists of three separate use cases: (a) its direct application in making sensory modality partitions of Wikidata; (b) validating and extending MultiMO classes and properties using LLMs; and (c) reverse engineering MultiMO resources to create a collection of competency questions on multisensory and multimodal content.

A Use Case on Wikidata Multimedia and Disability Items

In this use case, we study Wikidata as a target knowledge graph that could be annotated with MultiMO properties. A precondition for this is to study the multimedia content currently existing in Wikidata, as well as how this multimedia content has evolved over time [42].

This involves collecting multimedia content items' data from Wikidata using API, which will be enhanced with additional attributes such as timestamps, editor activity logs, and unique identifiers (QIDs) for multimedia content items. We also collect the usernames of editors that were involved in the creation of such multimedia content. These could be useful for future studies of multimodality in e.g. examining whether there are small-world networks developing among the editors, and to understand their community structure.

Figures YY and ZZ show the current partitions of Wikidata in terms of number of items containing images, audio and video content, as well as the evolution of these multimedia types and the number of editors supporting them. Overall, image records are the multimedia type occurring most frequently, closely followed by video and audio. Edits over the last two years have seen a decline in contributions for all three media types.

To further evaluate MultiMO, these partitions can be assigned to new *mmo:SensoryModality* subsets and refined accordingly to various specifications of sensory ability according to the *mmo:supportsSense* and *mmo:ICFCategory* and *mmo:ICFQualifier*. Specific use cases of users of Wikidata with different access

needs and recognised disabilities that match the ICF classification could be useful to test the usefulness of such partitions.

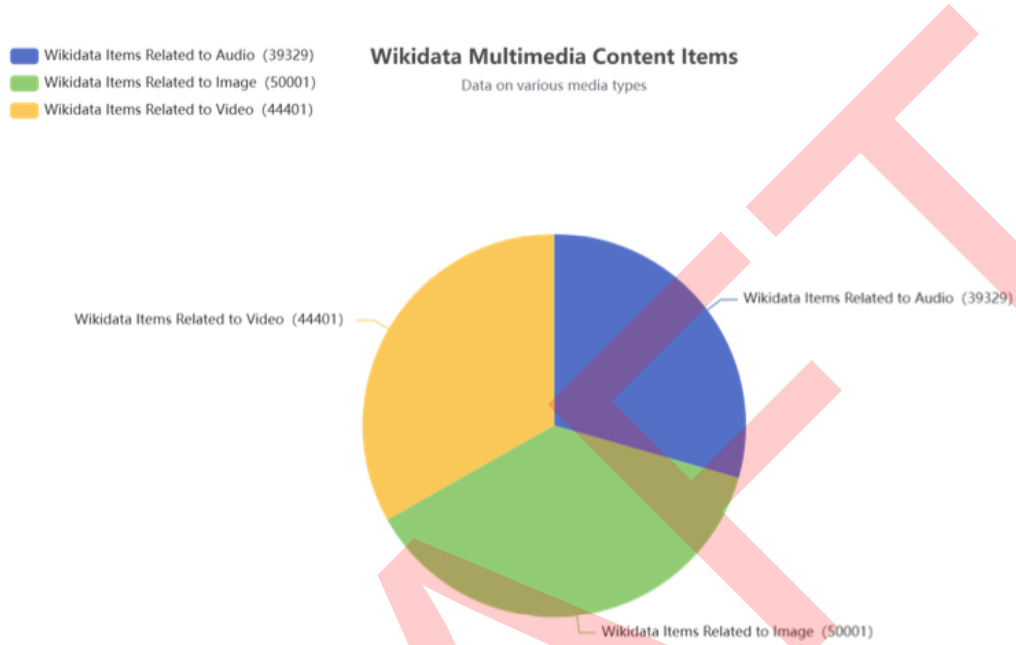

Figure 8: Distribution of audio, image, and video content in Wikidata items (from [42]).

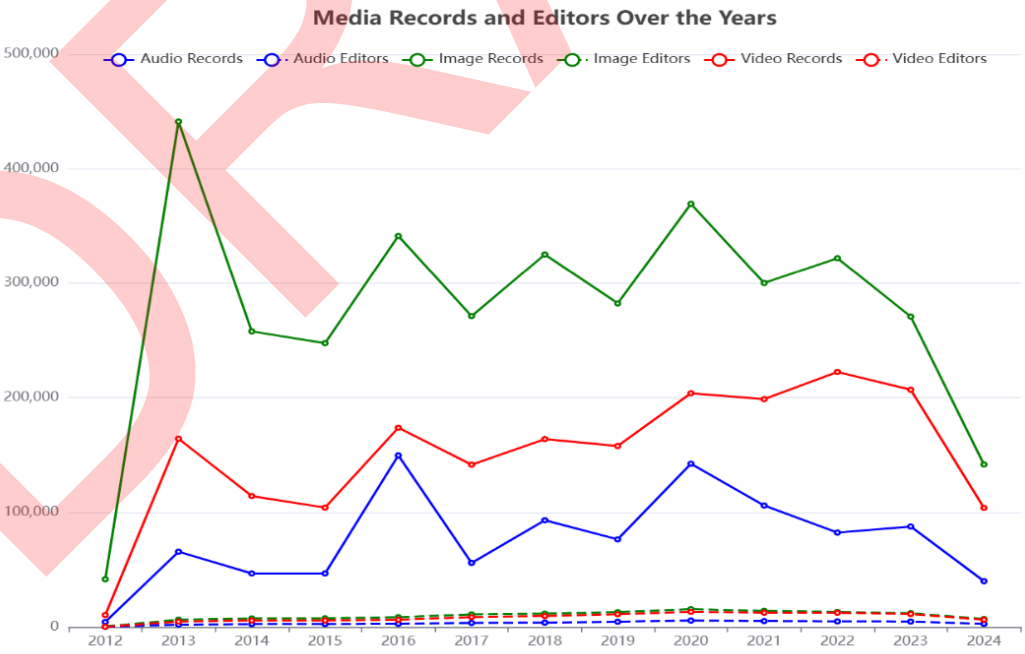

Figure 9: Distribution of audio, image, and video content in Wikidata over time (from [42]).

Extending MultiMO with LLMs and OntoChat

Following on our previous research on knowledge engineering with large language models (LLMKE), we propose a straightforward approach of continuing the modelling of MultiMO by using generative AI and LLMs.

In this case, we would like to collect user stories about the requirements of designing ontologies and vocabularies supporting multimodality and multisensory input. The idea is, through these user stories, to lift high quality requirements about desired features for such ontologies and vocabularies, beyond those we propose here. This could act in tandem with the previous use case, in which final users of Wikidata and other knowledge graphs with access needs and specific recognised disabilities interact with Ontochat to provide their user stories. Then, in a conversational setting, Ontochat can help them in fine-tuning their requirements and turn them into competency questions that can be used to add, change, or delete features (classes, properties) in MultiMO.

Reverse Engineering of Multisensory Competency Questions with RevOnt

Another possibility is to follow a backwards design approach, in which MultiMO features are first conceptualized and implemented (as we propose here), and then are reverse engineered and turned into competency questions. The advantage of doing so is that the use of MultiMO can start right away, and users can contribute and collaborate in a co-design approach with a (minimally) working vocabulary. Then, LLMs can be used to infer what competency questions can possibly be answered by MultiMO's classes and properties in its current state.

To follow this approach, we plan to use RevOnt [15], a state-of-the-art method that takes existing knowledge graphs as input, and reverse engineers them to propose competency questions that fit them. Similarly to the previous use case, we intend to do this in combination with OntoChat, in order to enrich the vocabulary that currently exists but also allowing users to provide their input and user stories to refine the competency questions. In this way, we can ensure to merge a top-down design (as we propose here) in which ontology elements are created from pre-existing requirements and competency questions; with a bottom-up design that leverages existing specifications and user input to establish and refine requirements.

5.5 Conclusion

Here, we propose MultiMO, a vocabulary for documenting sensory modalities in knowledge graphs. The main tenet of MultiMO is to combine existing vocabularies, such as DCAT and VoID, and extend them to support an abstraction that merges multisensory requirements and multimodal representations. This allows users and applications to directly query the knowledge graph for content that can be consumed by users with specific disabilities and access requirements that conform with WHO standards like ICF.

In the future, we will further refine MultiMO to allow the specification of more media content, and to expand the use of *dcat:mediaType* into a more complex class to specify multimodality. Second, we plan to represent ICF and other taxonomies that represent disability in a standard form that can be directly used in knowledge graphs, incorporating historical views on disability. Finally, we will integrate MultiMO

with other related efforts in documenting multimodality and multisensory information in open data and machine learning datasets, such as Croissant;[7] [36] and deploy it in various datasets beyond Wikidata.

5. Overall Conclusions

In conclusion, the ongoing transformation of ontology engineering through the integration of generative AI represents a promising shift towards more efficient and scalable knowledge management. By leveraging AI techniques such as LLMKE and OntoChat, the MuseIT project is pioneering new methods for developing and refining ontologies that can address the complex demands of multisensory and multimodal datasets. The creation and evaluation of the MultiMO ontology not only demonstrates the potential of these technologies but also highlights the importance of maintaining a collaborative human-AI approach. This synergy ensures that the resulting ontologies are both technically robust and contextually relevant, ultimately advancing the accessibility and equity of knowledge across diverse domains.

References

[1] Rabbit. 2023. Learning human actions on computer applications. https://www.rabbit.tech/research Accessed on January, 2024.

[2] Bohui Zhang, Albert Meroño Peñuela, and Elena Simperl. 2023. Towards Explainable Automatic Knowledge Graph Construction with Human-in-the-Loop. In HHAI 2023: Augmenting Human Intellect. IOS Press, Munich, Germany, 274–289.

[3] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models.

[4] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In International Conference on Machine Learning. PMLR, International Conference onMachine Learning (ICML), Honolulu, Hawaii, USA, 15696–15707.

[5] Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In Findings of the Association for Computational Linguistics: EMNLP 2021. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2370–2381.

[6] Abhijeet Kumar, Abhishek Pandey, Rohit Gadia, and Mridul Mishra. 2020. Building knowledge graph using pre-trained language model for learning entity-aware relationships. In 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON). IEEE, IEEE, Greater Noida, India, 310–315.

[7] Yifan Liu, Bin Shang, Chenxin Wang, and Yinliang Zhao. 2023. Knowledge Graph Completion with Information Adaptation and Refinement. In International Conference on Advanced Data Mining and Applications. Springer, Springer, Cham, Shenyang, China, 16–31.

[8] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. Artificial Intelligence Review 56 (2023),1–32.

[9] Rui Zhang, Yixin Su, Bayu Distiawan Trisedya, Xiaoyan Zhao, Min Yang, Hong Cheng, and Jianzhong Qi. 2023. AutoAlign: Fully Automatic and Effective Knowledge Graph Alignment enabled by Large Language Models. IEEE Transactions on Knowledge and Data Engineering Early Access (2023), 1–14.

[7] https://mlcommons.org/working-groups/data/croissant/

[10] Nicola Guarino and Christopher A Welty. 2009. An overview of OntoClean. ,201–220 pages.
[11] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. 2014. Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. International Journal on Semantic Web and Information Systems (IJSWIS) 10, 2 (2014), 7–34.
[12] Zhang B, Carriero VA, Schreiberhuber K, Tsaneva S, Gonzalez LS, Kim J, et al. OntoChat: a Framework´ for Conversational Ontology Engineering using Language Models. arXiv preprint arXiv:240305921. 2024.

[13] de Berardinis J, Carriero VA, Jain N, Lazzari N, Merono-Penuela A, Poltronieri A, et al. The Polifonia Ontology Network: Building a Semantic Backbone for Musical Heritage. In: Proceedings of the 22nd International Semantic Web Conference (ISWC); 2023.

[14] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Advances in neural information processing systems. 2020; 33:1877-901.

[15] Ciroku F, de Berardinis J, Kim J, Merono-Penuela A, Presutti V, Simperl E. RevOnt: Reverse Engineer-˜ ing of Competency Questions from Knowledge Graphs via Language Models. Manuscript under review. 2024.

[16] Aharoni R, Goldberg Y. Unsupervised Domain Clusters in Pretrained Language Models. In: Jurafsky D, Chai J, Schluter N, Tetreault J, editors. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics; 2020. p. 7747-63.

[17] Zhang Y, Wang Z, Shang J. ClusterLLM: Large Language Models as a Guide for Text Clustering. In: Bouamor H, Pino J, Bali K, editors. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics; 2023. p. 1390320.

[18] Viswanathan V, Gashteovski K, Lawrence C, Wu T, Neubig G. Large Language Models Enable FewShot Clustering. arXiv preprint arXiv:230700524. 2023.

[19] Blomqvist E, Seil Sepour A, Presutti V. Ontology testing-methodology and tool. In: Knowledge Engineering and Knowledge Management: 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings 18. Springer; 2012. p. 216-26.

[20] de Berardinis J, Carriero VA, Merono-Penuela A, Poltronieri A, Presutti V. The Music Meta Ontology: ˜ a flexible semantic model for the interoperability of music metadata. In: Proceedings of the 24th International Society for Music Information Retrieval Conference; 2023.

[21] de Berardinis J, Penuela AM, Jain N, Poltronieri A, Lazzari N, Presutti V, et al. Ontologies and knowledge graphs of music objects, patterns, and software package – 2nd version. European Commission, The Polifonia consortium; 2023.

[22] Sarkar A, Drosos I, Deline R, Gordon AD, Negreanu C, Rintel S, et al. Participatory prompting: a user-centric research method for eliciting AI assistance opportunities in knowledge workflows; 2023. Available from: https://arxiv.org/abs/2312.16633.

[23] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, et al., A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity, arXiv preprint arXiv:2302.04023 (2023).

[24] OpenAI, GPT-4 Technical Report, 2023. arXiv:2303.08774.

[25] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., LLaMA: Open and Efficient Foundation Language Models, arXiv preprint arXiv:2302.13971 (2023).

[26] Z. Li, Z. Yang, M. Wang, Reinforcement Learning with Human Feedback: Learning Dynamic Choices via Pessimism, 2023. arXiv:2305.18438.

[27] G. Qin, J. Eisner, Learning how to ask: Querying LMs with mixtures of soft prompts, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 5203–5212. URL: https://aclanthology.org/2021.naacl-main.410. doi:10.18653/v1/2021.naacl-main.410.

[28] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts, arXiv preprint arXiv:2010.15980 (2020).

[29] D. Alivanistos, S. B. Santamaría, M. Cochez, J. C. Kalo, E. van Krieken, T. Thanapalasingam, Prompting as Probing: Using Language Models for Knowledge Base Construction, in: 2022 Semantic Web Challenge on Knowledge Base Construction from Pre-Trained Language Models, LM-KBC 2022, CEUR-WS. org, 2022, pp. 11–34.

[30] S. Singhania, T.-P. Nguyen, S. Razniewski, LM-KBC: Knowledge base construction from pre-trained language models, 2022.

[31] F. Petroni, P. S. H. Lewis, A. Piktus, T. Rocktäschel, Y. Wu, A. H. Miller, S. Riedel, How Context Affects Language Models' Factual Predictions, in: D. Das, H. Hajishirzi, A. McCallum, S. Singh (Eds.), Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020, 2020. URL: https://doi.org/10.24432/C5201W. doi:10.24432/C5201W.

[32] B. Zhang, I. Reklos, N. Jain, A.M. Peñuela, E. Simperl, Using Large Language Models for Knowledge Engineering (LLMKE): A Case Study on Wikidata, arXiv preprint arXiv:2309.08491 (2023).

[33] Zhao Y, Zhang B, Hu X, Ouyang S, Kim J, Jain N, de Berardinis J, Meroño-Peñuela A, Simperl E. Improving Ontology Requirements Engineering with OntoChat and Participatory Prompting. arXiv preprint arXiv:2408.15256. 2024 Aug 9.

[34] Kannen, Nithish, et al. "Beyond Aesthetics: Cultural Competence in Text-to-Image Models." *arXiv preprint arXiv:2407.06863* (2024).

[35] Zhitomirsky-Geffet, M., Kizhner, I. and Minster, S. (2023), "What do they make us see: a comparative study of cultural bias in online databases of two large museums", *Journal of Documentation*, Vol. 79 No. 2, pp. 320-340. https://doi.org/10.1108/JD-02-2022-0047.

[36] Mubashara Akhtar, Omar Benjelloun, Costanza Conforti, Pieter Gijsbers, Joan Giner-Miguelez, Nitisha Jain, Michael Kuchnik, Quentin Lhoest, Pierre Marcenac, Manil Maskey, Peter Mattson, Luis Oala, Pierre Ruyssen, Rajat Shinde, Elena Simperl, Goeffry Thomas, Slava Tykhonov, Joaquin Vanschoren, Jos van der Velde, Steffen Vogler, and Carole-Jean Wu. Croissant: A metadata format for ml-ready datasets. In Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning, DEEM '24, page 1–6, New York, NY, USA, 2024. Association for Computing Machinery.

[37] Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., & Vrandečić, D. (2014). Introducing wikidata to the linked data web. In *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13* (pp. 50-65). Springer International Publishing.

[38] Hogan A, Blomqvist E, Cochez M, d'Amato C, Melo GD, Gutierrez C, Kirrane S, Gayo JE, Navigli R, Neumaier S, Ngomo AC. Knowledge graphs. ACM Computing Surveys (Csur). 2021 Jul 2;54(4):1-37.

[39] World Wide Web Consortium. "Data catalog vocabulary (DCAT)." (2014).

[40] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets. LDOW2009, 2009.

[41] Weibel SL, Koch T. The Dublin core metadata initiative. D-lib magazine. 2000 Dec;6(12):1082-9873.

[42] Xu, Chen. Extension of Dataset and Analysis of Multimedia Gaps. MSc Thesis, King's College London. 2024