Can LLMs Generate Competency Questions?

Youssra Rebboud[1] [0000-0003-3507-5646], Lionel Tailhardat[1,2] [0000-0001-5887-899X], Pasquale Lisena[1] [0000-0003-3094-5585], and Raphael Troncy[1] [0000-0003-0457-1436]

[1] EURECOM, Sophia Antipolis, France
[2] Orange, France

Abstract. Large Language Models have shown high performances in a large number of tasks, being recently applied also to support Knowledge Graphs construction. An important step for data modeling consists in the definition of a set of competency questions, which are often used as a guide for the development of an ontology and as a mean to evaluate the resulting schema. In this work, we investigate the suitability of LLMs for the automatic generation of competency questions given an existing ontology. We compare different large language models under various settings in order to give a comprehensive overview of what LLMs can do to support the knowledge engineer.

Keywords: LLMs · Knowledge Graphs · Ontology · Data Modeling

1 Introduction

Ontologies – as explicit representations of a discourse domain through concepts and relationships – and their instantiation as knowledge graphs, enable data analysis and inference techniques to handle heterogeneous data and reason about the context of represented objects. Despite these advantages and proven knowledge engineering methods [15,17], designing an ontology represents a significant upfront cost for application designers willing to build and leverage a knowledge graph. Indeed, modeling an application domain requires knowledge engineers to immerse themselves in the domain over a long period of time and engage with numerous domain experts. Simultaneously, the recent explosive success of generative AI methods and the widespread use of large language models (LLMs) as a crucial component in industrial and consumer applications – particularly in the field of generating code from user-expressed intentions in natural language (and vice versa) – suggests that abstracting a domain into a specific formalism from a textual corpus is an achievable goal that could assist knowledge engineers in their work. Simple experiments, accessible to anyone via ChatGPT or similar tools, demonstrate that it is indeed possible to generate a skeleton of an OWL/RDF ontology implementation using a prompt that briefly describes the targeted concepts. However, the reliability and scalability of this intuition still need to be explored, which leads us to ask the question of how much LLMs could co-contribute in the knowledge engineering process together with usual knowledge engineering methodologies (competency questions, ontology re-use, authoring tests, etc.).

In order to thoroughly explore the intricacies of this question, we have identified six sub-tasks, which are presented in Table 1. In this paper, we delve into the details of the sub-task #1, assuming that insights gained from this research will likely contribute to advancements in the other sub-tasks. Our approach consists in analyzing the quality of the Competency Questions (CQs) [17] generated by LLMs through prompt-engineering experiments. These experiments are conducted on a dataset of RDF-based ontologies, along with their corresponding set of CQs and evaluation queries provided by the authors of each ontology. Through this work, we contribute to boosting the adoption of Semantic Web technologies and research on LLMs by defining a methodology for exploring the coupling of LLMs and Knowledge Graphs (KGs) with a focus on ontologies. We also highlight which ontology characteristics or LLM parameter settings are crucial in facilitating knowledge engineering tasks. The dataset and code related to this work is available at `https://github.com/D2KLab/llm4ke`.

Table 1. 6 sub-tasks essential in the knowledge engineering process

#	Research questions – Could a LLM ...
1	reverse engineer an ontology and find out what good competency questions (CQ) could be derived?
2	take as input the CQ and generate parts of the ontology?
3	take as input the CQ and extend an existing ontology?
4	take as input the CQ and generate ontology design patterns?
5	write an authoring test (a SPARQL query) given the ontology and the CQ?
6	generate an adequate set of RML rules for data ingestion given a dataset and an ontology?

The remainder of this paper is organized as follows. In Section 2, we review some related work. In Section 3, we provide details of our approach by focusing on the LLM-based data processing pipeline and describing how to perform prompting. In Section 4, we present the experiments conducted and their results on a subset of five RDF ontologies and six LLMs. We conclude and outline some future work in Section 5.

2 Related Work

Different works have so far investigated the performance of LLMs in classic tasks in the Knowledge Graph domain [2]. *SPIRES* [6] is a method that utilizes GPT-3 to produce structured data from an input text and schema. In [20], the authors use the Overall Execution Accuracy (OEA) to assess the performance of a LLM in converting questions to queries (SQL or SPARQL). The OEA is computed on an ad-hoc benchmark, where an execution is considered accurate if the query result matches the corresponding answer.

Several works address the usage and production of competency questions. The study of patterns in competency questions [25] has inspired the realization of

AgOCQs [3] in which CQs are automatically generated. The evaluation has been performed with an expert group, which highlighted the validity of the method. The patterns can be filled by *Glossary of terms* – which can be automatically extracted such as in ReqTagger [26] – or used to automatically generate SPARQL queries from CQs [4,24].

3 Methodology

In this section, we provide details of our approach by focusing in Section 3.1 on the LLM-based data processing pipeline, and then describing in Section 3.2 the prompt details.

3.1 Implementation

To standardize and automate experiments, we developed a platform in Python, whose workflow is depicted in Figure 1. The platform relies on the LangChain framework[3] [7] to interact with various LLMs. Specifically, we integrated models from LangChain providers for Ollama, HuggingFace, and OpenAI into our workflow, allowing for querying within the same pipeline.

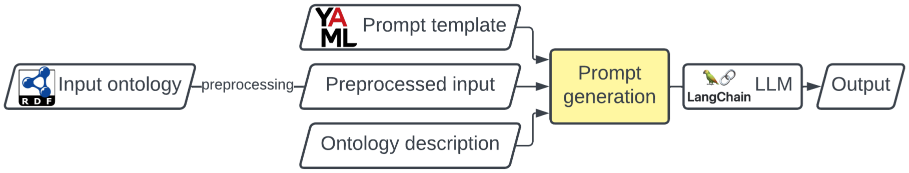

Fig. 1. Workflow of the platform

We make use of a prompt configuration in the form of a YAML file, including:

- the description of the task TD (for documentation purposes);
- the list of required input fields;
- the prompt template, in which the placeholders for input values are marked by curly brackets as in the documentation of LangChain, e.g. {`name`}, {`classes`}.

Additionally, each process can be further customized by specifying the LLM to use, the path of the input ontology, whether to include the ontology description in the prompt or not, and the number of required output results.

In order to avoid to ingest the full RDF representation in the prompt[4], we parse the ontology using RDFlib [11] and extract either:

[3] `https://python.langchain.com/`

[4] During some preliminary experiments, we realised that including the full ontology in Turtle format was producing a long prompt, which has shown to confuse the LLMs and produce hallucination.

- the list of class labels C;
- the list of property labels P;
- a summary schema of the interconnection of classes and properties S.

This schema S is represented as triples in the format (C_x, p_y, C_z), where $C_x, C_z \in C$ are class labels, and $p_y \in P$ is the label of an object property which has C_x as domain and C_z as range. An example taken from the FOAF ontology is `(foaf:Group, foaf:member, foaf:Agent)`. Please note that C_x and C_z are not necessarily two different classes, because the domain and range can coincide, e.g. in `(foaf:Person, foaf:knows, foaf:Person)`. In the case of a data property $p_d \in P$, we include the triple $(C_x, p_d, \text{“}literal\text{”})$, e.g. in `(foaf:Person, foaf:lastName, "literal")`.

When the dimension of the ontology is large, it is processed in batches of 20 classes. In such a case, in each iteration, C is composed of a maximum of 20 classes, P includes all properties which have C as domain or range, and S encompasses all interconnections involving C and P.

3.2 Prompting

We primarily utilized three templates for our work. The first template outlines the classes within the ontology, the second includes both classes and properties, and the final template integrates the ontology's schema. Each of these templates encompasses:

- Task Description (TD): `‘Generate a set of competency questions (CQ) which are relevant for the ontology called {name of ontology}’`.
- Ontology Description (OD): provides a general overview of the ontology and specifies the domain it belongs to, e.g., `‘Odeuropa ontology represents’ odours and their experiences from Cultural Heritage perspective`.
- Examples (EXP): examples of CQs of the desired ontology, e.g., `‘Which scents were linked to the idea of heaven in X period?’`.
- Notes(N): guidelines provided to the model for brevity and clarity, e.g., `‘Do not include any text except the competency question’`.

Based on the prompt configuration technique described in Section 3.1, we propose to generate prompts for a given ontology with various features (Table 2) depending on the overall experiment goals and following best practice in prompt structuring.

4 Experiments

In this section, we present the experiments conducted based on the method described in Section 3. We first provide details of the dataset used in Section 4.1, then on the LLMs used in Section 4.2, and finally report on the evaluation results in Section 4.3.

Table 2. Prompt features as a function of the evaluation goal.
For the classes feature, the “*The {name} ontology has the following set of classes:*” is used in the prompt. For the ”Properties” feature, it is the “*and the following set of properties:*” sentence. For “Schema” it is the “*The {name} ontology has the following schema*” sentence. “opt.” stands for optional (i.e. w. and w.o definition).

Evaluation goal	Definition	Classes	Properties	Schema	Examples	Constraints
All classes	opt.	✓			n	✓
All classes + properties	opt.	✓	✓		n	✓
Logic	opt.			✓	n	✓

4.1 Investigated Ontologies

For our experiments, we selected a subset of five ontologies (Table 3) with a publicly available implementation based on the following two criteria: *1)* these ontologies were modeled following explicitly the Competency Questions (CQs) methodology [17]; *2)* these ontologies have well-phrased CQs with associated Authoring Tests (ATs) in the form of SPARQL queries. Once the subset was established, we created a dataset by recording a versioned copy of the ontologies' implementation, as well as their companion set of CQs and ATs. To generalize the approach described in Section 3 to all the ontologies of the subset, we normalized the representation of the CQs by storing them in a YAML data structure including – if relevant – the reference to the corresponding ATs. The dataset is publicly available in our repository, with annotation on the origin for each component of it and explanations on the normalization process.

Table 3. Subset of ontologies for the LLM4KE experiments.
Ontologies in our dataset, along with additional details such as the number of classes (#Classes), associated competency questions (CQ count), associated authoring tests (AT count), and a coverage measure (AT/CQ coverage) indicating the extent to which ATs are effectively defined and implemented for each CQ. For Polifonia, we count CQs from their “default group” and indicate “?” for the AT count as no obvious set of ATs was found. For Demcare, the CQ2SPARQLOWL [14] dataset served as a reference for building our dataset. For the remaining ontologies, the dataset was directly constructed from each project's repository.

Data-model	Ref.	Full ontology name or topic	#Classes	CQ count	AT count	AT/CQ coverage
DemCare	[10]	Dementia Ambient Care Ontology.	290	107	60	56%
DOREMUS	[1]	Music catalogues on the web of data.	218	58	30	52%
NORIA-O	[21]	IT networks, events and operations for anomaly detection and IT service management.	55	26	25	88%
Odeuropa	[12]	Odours and their experiences from a Cultural Heritage perspective.	13	74	74	100%
Polifonia	[5]	Polifonia Ontology Network (PON) for queries in the music domain.	247	194	?	0%

4.2 Investigated LLMs

We have explored various Large Language Model (LLM) options for our experiments, including both open-source and proprietary models. For open-source models, we considered their performances according to the Hugging Face leaderboard,[5] in particular across three specific datasets, which we consider relevant for this research:

- ARC2018 [8] (AI2 Reasoning Challenge), a question-answering dataset;
- HellaSwag [27], a dataset created to challenge model common sense reasoning abilities;
- Winogrande [18], a dataset designed to evaluate commonsense reasoning capabilities in AI systems.

We selected these models based on their distinct architectures, aiming to choose one from each architectural category. Each model was chosen for its superior performance within its respective architecture, as indicated by their positions on the leaderboard at the time of selection. Due to resource limitations, we have opted to confine our selection of open-source LLMs to those with a parameter count equal to or less than 13 billion. Table 4 summarises the used LLMs.

Table 4. Used LLMs for Experiments. B refers to billion parameters.

Model	Architecture	Size (B)	Access Paradigm
DPO[6]	MixtralForCausalLM	12.9	Open-source
Solar[8]	LlamaForCausalLM	10.7	Open-source
UNA[10]	MistralForCausalLM	7	Open-source
Zephyrβ [12]	MistralForCausalLM	7	Open-source
GPT 3.5	Transformer Decoder	175	proprietary
GPT-4-0125-preview	Transformer Decoder	1500	proprietary

We have used `Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B`[6] (we refer to it as *DPO*), which is an instance of `FusionNet_7Bx2_MoE_14B` fine-tuned on the Truthy-DPO dataset[7].

Additionally, we leveraged `SOLAR-10B-OrcaDPO-Jawade`, which we shortcut to *Solar*, a finetuned version of `SOLAR-10.7B-Instruct-v1.0`[8] [9], finetuned on the `dpo pairs` dataset.[9] Furthermore, we have used `UNA-TheBeagle-7b-v1`[10], that we call simply *UNA*, a 7B LLM trained on The Bagel dataset.[11] On the other hand, we opted for zephyrβ[12] [23], because of its performance that surpassed Llama2 70B [22] on different benchmarks.

[5] `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`

[6] `https://huggingface.co/yunconglong/Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B`

[7] `https://huggingface.co/datasets/jondurbin/truthy-dpo-v0.1`

[8] `https://huggingface.co/bhavinjawade/SOLAR/-10B/-OrcaDPO/-Jawade`

[9] `https://huggingface.co/datasets/Intel/orca_dpo_pairs`

[10] `https://huggingface.co/fblgit/UNA-TheBeagle/-7b/-v1`

[11] `https://huggingface.co/datasets/jondurbin/bagel-v0.3`

[12] `https://huggingface.co/HuggingFaceH4/zephyr/-7b/-beta`

Moreover, we included in our study API-only access models, and in particular the GPT series from OpenAI[13]. We used both GPT3.5[14] and GPT4 [13].

4.3 Evaluation

Comparative analysis of generated CQs with ground-truth. To perform the evaluation of our approach, we utilize the dataset presented in Section 4.1 and consider the CQs provided by the authors of each ontology as the ground truth. We compare the output CQs from the LLMs (CQ_o) to each CQ in the ground-truth (CQ_{gt}) and consider a CQ_o as valid if it is sufficiently similar to at least one CQ_{gt}. For the similarity score, we use cosine similarity between the embeddings of CQ_o and CQ_{gt} computed using SentenceBERT [16]. We define a threshold θ above which we consider a CQ_o valid (Eq. 1):

$$x \in CQ_o^{valid} \Leftrightarrow x \in CQ_o \wedge \exists \{y \in CQ_{gt} : \text{cosine similarity}(y, x) > \theta\} \qquad (1)$$

with $CQ_o^{valid} \subset CQ_o$. We then compute the precision $P = \frac{\text{number of } CQ_o^{valid}}{\text{number of } CQ_o}$ of each experiment.

Results & discussion. The results of the experiments are reported in Table 5, using a threshold of $\theta = 0.6$, chosen empirically for better showing the differences between the models. As a first outlook, we observe that the precision scores are generally low. From the perspective of the LLMs, Zephyr consistently shows the best scores across a majority of ontologies, with the exception of Odeuropa. This can be due to the fact that the dimension of Odeuropa is lower that the used batch size, and it is consequently included entirely in the prompt; reducing the batch size to 5, improves the results of Zephyr for Odeuropa to 0.90 (C), 0.91 (P) and 0.70 (S). Future work will investigate the effect of the batch size on the different LLMs and ontologies.

From the perspective of prompt features, we observe that providing examples (few-shot) generally leads to better precision (compared to zero-shot), although not always. Similarly, using properties in prompts results in a greater increase in precision. Conversely, prompting with the schema does not generally improve precision and may even decrease it, as in the case of GPT4, DPO and Zephyr.

Even though the absolute scores are generally quite low, it should not be concluded that the generated CQs are irrelevant. In fact, the generation process may have resulted in new competency questions that can be a valuable addition to the ground truth dataset. To properly evaluate the relevance of these competency questions, an expert panel should be involved, which will be the focus of future work. Due to variations in the number of classes among the ontologies in our dataset (Table 3), it is important to note that the LLMs used in the experiments may have been queried more frequently for certain ontologies and less frequently for others, because of the subdivision in batches.

[13] `https://openai.com/`

[14] `https://platform.openai.com/docs/models/gpt-3-5-turbo`

Table 5. The precision scores for the experiments, reporting the LLM name, the number of included exemplary CQs and, for each ontology, the modality {C = all classes, P = classes and properties, S = summary schema}

Ontology →		DOREMUS			DemCare			Odeuropa			Polifonia			noria-o		
LLM	**Ex**	C	P	S	C	P	S	C	P	S	C	P	S	C	P	S
GPT3	0	0.02	0.01	0.01	0.15	0.14	0.00	0.00	0.00	0.10	0.08	0.08	0.20	0.00	0.00	0.03
	3	0.04	0.01	0.04	0.17	0.13	0.00	0.90	0.30	0.00	0.20	0.30	0.32	0.00	0.03	0.03
GPT4	0	0.00	0.00	0.02	0.14	0.23	0.01	0.20	0.50	0.30	0.21	0.24	0.30	0.00	0.03	0.00
	3	**0.10**	0.11	0.11	**0.21**	0.17	0.01	0.40	0.90	0.90	0.32	0.32	0.32	0.03	0.03	0.00
dpo	0	0.00	0.00	0.00	0.04	0.08	0.00	0.70	0.30	0.00	0.05	0.09	0.11	0.00	0.00	0.00
	3	0.03	0.04	0.01	0.15	0.13	0.04	0.75	0.82	**1.00**	0.22	0.22	0.22	0.04	0.06	0.00
solar	0	0.00	0.00	0.00	0.08	0.06	0.00	0.20	0.00	0.20	0.07	0.04	0.12	0.00	0.03	0.00
	3	0.00	0.12	0.07	0.11	0.17	0.00	0.30	0.30	0.30	0.20	0.22	0.24	0.04	0.00	0.03
una	0	0.00	0.03	0.05	0.10	0.10	0.00	0.50	0.00	0.64	0.08	0.05	0.10	0.03	0.00	0.00
	3	0.09	0.15	0.12	0.20	0.24	**0.27**	**1.00**	0.70	**1.00**	0.34	**0.38**	0.33	**0.31**	**0.07**	0.00
zephyr	0	0.01	0.01	0.00	0.05	0.09	0.00	0.90	**1.00**	0.00	0.16	0.08	0.15	0.00	0.00	0.00
	3	0.03	**0.58**	**0.56**	**0.21**	**0.33**	0.00	0.40	0.00	**1.00**	**0.36**	**0.38**	**0.34**	0.00	0.00	**0.20**

5 Conclusion and Future Work

This work aimed to understand how knowledge engineering can benefit from large language models (LLMs). We identified six sub-tasks and developed a methodology to explore the coupling of LLMs with knowledge graphs, specifically focusing on ontologies. Using a data processing pipeline with six LLMs, three prompting strategies, and five ontologies, we assessed the ability of LLMs to generate competency questions, which are crucial in ontology development. In conclusion, providing examples of competency questions and utilizing relationship information from ontologies in prompts is important for improving LLM performance. It is interesting to note that providing more details for certain ontologies can decrease LLM performance, which requires further investigation.

Future work will focus on understanding the characteristics of ontologies that impact the accuracy of LLM responses. This includes investigating the relevance of LLMs trained on general language for ontologies with specialized vocabulary. Additionally, research will explore the role of Competency Question formulation and the influence of properties, including their names, descriptions, and associated logic. Evaluating the capability of LLMs to handle ontologies that reuse other data models will also be explored. To provide more generalizable results, the work will be extended to other ontologies with well-formulated Competency Questions and Authoring Tests, such as using the CQ2SPARQLOWL dataset [14] and the SILKNOW ontology [19].

References

1. Achichi, M., Lisena, P., Todorov, K., Troncy, R., Delahousse, J.: DOREMUS: A Graph of Linked Musical Works. In: 17^{th} International Semantic Web Conference (ISWC). Monterey, CA, USA (10 2018)
2. Allen, B., Stork, L., Groth, P.: Knowledge Engineering Using Large Language Models. Transactions on Graph Data and Knowledge (2023). https://doi.org/10.4230/TGDK.1.1.3, `https://drops.dagstuhl.de/entities/document/10.4230/TGDK.1.1.3`

3. Antia, M.J., Keet, C.M.: Automating the Generation of Competency Questions for Ontologies with AgOCQs. In: Ortiz-Rodriguez, F., Villazón-Terrazas, B., Tiwari, S., Bobed, C. (eds.) Knowledge Graphs and Semantic Web. pp. 213–227. Springer Nature Switzerland, Cham (2023)
4. Benhocine, K., Hansali, A., Zemmouchi-Ghomari, L., Ghomari, A.R.: Towards an automatic SPARQL query generation from ontology competency questions. International Journal of Computers and Applications **44**(10), 971–980 (2022). https://doi.org/10.1080/1206212X.2022.2031722
5. de Berardinis, J., Carriero, V.A., Jain, N., Lazzari, N., Meroño-Peñuela, A., Poltronieri, A., Presutti, V.: The Polifonia Ontology Network: Building a Semantic Backbone for Musical Heritage. In: The Semantic Web – ISWC 2023 (2023)
6. Caufield, J.H., Hegde, H., Emonet, V., Harris, N.L., Joachimiak, M.P., Matentzoglu, N., Kim, H., Moxon, S., Reese, J.T., Haendel, M.A., Robinson, P.N., Mungall, C.J.: Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES): a method for populating knowledge bases using zero-shot learning. Bioinformatics p. 104 (02 2024). https://doi.org/10.1093/bioinformatics/btae104
7. Chase, H.: LangChain (Oct 2022), https://github.com/langchain-ai/langchain
8. Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., Tafjord, O.: Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. ArXiv **abs/1803.05457** (2018), https://api.semanticscholar.org/CorpusID:3922816
9. Kim, D., Park, C., Kim, S., Lee, W., Song, W., Kim, Y., Kim, H., Kim, Y., Lee, H., Kim, J., Ahn, C., Yang, S., Lee, S., Park, H., Gim, G., Cha, M., Lee, H., Kim, S.: SOLAR 10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling (2023)
10. Kompatsiaris, I.: Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support. https://demcare.eu/ (2012)
11. Krech, D., Grimnes, G.A., Higgins, G., Hees, J., Aucamp, I., Lindström, N., Arndt, N., Sommer, A., Chuc, E., Herman, I., Nelson, A., McCusker, J., Gillespie, T., Kluyver, T., Ludwig, F., Champin, P.A., Watts, M., Holzer, U., Summers, E., Morriss, W., Winston, D., Perttula, D., Kovacevic, F., Chateauneu, R., Solbrig, H., Cogrel, B., Stuart, V.: RDFLib (Aug 2023). https://doi.org/10.5281/zenodo.6845245, https://github.com/RDFLib/rdflib
12. Lisena, P., Schwabe, D., van Erp, M., Troncy, R., Tullett, W., Leemans, I., Marx, L., Ehrich, S.C.: Capturing the Semantics of Smell: The Odeuropa Data Model for Olfactory Heritage Information. In: The Semantic Web. pp. 387–405. Springer International Publishing, Cham (2022)
13. OpenAI: GPT-4 Technical Report (2023)
14. Potoniec, J., Wiśniewski, D., Ławrynowicz, A., Keet, C.M.: Dataset of ontology competency questions to SPARQL-OWL queries translations. Data in Brief (2020). https://doi.org/10.1016/j.dib.2019.105098
15. Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., García-Castro, R.: LOT: An industrial oriented ontology engineering framework. Engineering Applications of Artificial Intelligence. Engineering Applications of Artificial Intelligence **111**, 104755 (2022). https://doi.org/10.1016/j.engappai.2022.104755
16. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019)

17. Ren, Y., Parvizi, A., Mellish, C., Pan, J., van Deemter, K., Stevens, R.: Towards Competency Question-Driven Ontology Authoring. In: 11^{th} European Semantic Web Conference (ESWC) (2014). https://doi.org/10.1007/978-3-319-07443-6_50
18. Sakaguchi, K., Bras, R.L., Bhagavatula, C., Choi, Y.: WinoGrande: an adversarial winograd schema challenge at scale. Commun. ACM **64**(9), 99–106 (aug 2021). https://doi.org/10.1145/3474381
19. Schleider, T., Troncy, R., Gaitan, M., Alba, E., et al.: The SILKNOW Knowledge Graph. Semantic Web Journal, Special Issue on Cultural Heritage and Semantic Web, March 2021, IOS Press (2021)
20. Sequeda, J., Allemang, D., Jacob, B.: A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases (2023)
21. Tailhardat, L., Chabot, Y., Troncy, R.: NORIA-O: an Ontology for Anomaly Detection and Incident Management in ICT Systems. In: Semantic Web – 21^{st} International Conference, ESWC 2024, Hersonissos, Crete, Greece, May 26 - 30, 2024, Proceedings (2024)
22. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M.A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T.: Llama 2: Open Foundation and Fine-Tuned Chat Models (2023)
23. Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A.M., Wolf, T.: Zephyr: Direct Distillation of LM Alignment (2023)
24. Wisniewski, D., Potoniec, J., Lawrynowicz, A.: SeeQuery: An Automatic Method for Recommending Translations of Ontology Competency Questions into SPARQL-OWL. In: 30th ACM International Conference on Information & Knowledge Management (CIKM). p. 2119–2128. Association for Computing Machinery (2021). https://doi.org/10.1145/3459637.3482387
25. Wisniewski, D., Potoniec, J., Lawrynowicz, A., Keet, C.M.: Competency Questions and SPARQL-OWL Queries Dataset and Analysis. Journal of Web Semantics **59**, 100534 (2019). https://doi.org/10.1016/j.websem.2019.100534, https://www.sciencedirect.com/science/article/pii/S1570826819300617
26. Wiśniewski, D., Potoniec, J., Ławrynowicz, A.: ReqTagger: A Rule-Based Tagger for Automatic Glossary of Terms Extraction from Ontology Requirements. Foundations of Computing and Decision Sciences **47**(1), 65–86 (2022). https://doi.org/10.2478/fcds-2022-0003
27. Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y.: HellaSwag: Can a Machine Really Finish Your Sentence? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)