Developing a Scalable Benchmark for Assessing Large Language Models in Knowledge Graph Engineering

Lars-Peter Meyer[1,2,3,*], Johannes Frey[1,2,3,†], Kurt Junghanns[1,2,†], Felix Brei[1,†], Kirill Bulert[1], Sabine Gründer-Fahrer[1,2] and Michael Martin[1,2]

[1]Institute for Applied Informatics, Goerdelerring 9, 04109 Leipzig, Germany, https://infai.org
[2]Agile Knowledge Engineering and Semantic Web (AKSW), https://aksw.org
[3]Leipzig University, Institute for Informatics, Germany, https://www.uni-leipzig.de

Abstract

As the field of Large Language Models (LLMs) evolves at an accelerated pace, the critical need to assess and monitor their performance emerges. We introduce a benchmarking framework focused on knowledge graph engineering (KGE) accompanied by three challenges addressing syntax and error correction, facts extraction and dataset generation. We show that while being a useful tool, LLMs are yet unfit to assist in knowledge graph generation with zero-shot prompting. Consequently, our *LLM-KG-Bench* framework provides automatic evaluation and storage of LLM responses as well as statistical data and visualization tools to support tracking of prompt engineering and model performance.

Keywords

Large Language Model, Knowledge Graph Engineering, Large Language Model Benchmark

1. Introduction

Large Language Models (LLMs) hold the potential to change the way how we interact with data and technology. Especially models like GPT-3 and GPT-4 have shown proficient capabilities in solving textual assignments [1] and spawned a wave of subsequent models and the field of *prompt engineering*.

But the fast evolution and rapidly growing landscape of different LLMs make it challenging to keep track of their individual capabilities and to choose the best model and best prompt for the job. There exist efforts on generic LLM benchmarks (e.g. [2]). However, despite these advancements, the application and (automated) assessment of LLMs in the context of knowledge graph engineering (KGE) and the Semantic Web is still a highly under-explored area. In response to this gap, this paper proposes a first LLM KGE benchmarking framework *LLM-KG-Bench*[1] that follows our vision of an automated and continuous evaluation platform for different tasks

SEMANTICS 2023 EU: 19th International Conference on Semantic Systems, September 20-22, 2023, Leipzig, Germany
*Corresponding author.
†These authors contributed equally.
lpmeyer@infai.org (L. Meyer)
0000-0001-5260-5181 (L. Meyer); 0000-0003-3127-0815 (J. Frey); 0000-0003-1337-2770 (K. Junghanns); 0009-0008-5245-6655 (F. Brei); 0000-0002-1459-3754 (K. Bulert); 0000-0003-0054-5003 (S. Gründer-Fahrer); 0000-0003-0762-8688 (M. Martin)

[1]Repository: https://github.com/AKSW/LLM-KG-Bench or doi:10.5281/zenodo.8251944

in KGE scenarios. A test of the framework is presented by comparing three LLMs for three exemplary KGE tasks.

2. Related Work

The utilization of an LLM in the semantic web domain benefits from its capability to handle RDF-related syntaxes such as JSON-LD, Turtle and SPARQL. A comprehensive amalgamation of LLMs and knowledge graphs (KGs) is described in *Dagstuhl Seminar* [3] and [4]. The *Knowledge Base Construction from Pre-trained Language Models (LM-KBC) Challenge*[2] emphasises the relevance of this combination.

The basis of this study is [5], where ChatGPT's use in knowledge graph engineering is assessed. Impressive capabilities were revealed, suggesting two conclusions: Firstly, such studies offer insight into LLMs' potential and limitations, aiding knowledge graph engineers. Secondly, comparing different LLMs can lead to superior results by addressing inherent model issues.

Recognizing the potential of Large Language Models (LLMs) in knowledge graph engineering, it's vital to evaluate their performance across diverse tasks. Google's *Beyond the Imitation Game (BIG-bench) Benchmark*[3] [2] and the Large Model Systems (LMSys) leaderboard[4] are community efforts that assess the performance of various models with regard to a plethora of tasks. The *Language Model Evaluation Harness*[5] offers further testing of generative language models on various evaluation tasks. However all of them are not perfect for assessing an LLM's use for KGE. They are missing KGE specific scoring and do not evaluate scores relative to problem size. The size seems to be relevant for KGE as KGs get quite big in relation to current LLMs context sizes[5]. Acknowledging the existing appraoches limitations we introduce the *LLM-KG-Bench* framework.

3. The LLM-KG-Bench Framework

Our current (and ongoing) work presented in this paper is comprising the design and implementation of the modular *LLM-KG-Bench* framework[1] for benchmarking LLMs in the context of knowledge graph engineering. The main focus is on automated evaluation procedures to allow for many repeated test executions. The framework supports configurable task sizing, as prior work[5] suggest the relevance of the LLM's context size for KGE tasks.

As we aim for as much compatibility as possible, especially in the direction of *BIG-bench*[3], the *LLM-KG-Bench* framework is organized around *benchmark tasks* and *LLM model connectors*, glued together by some code for execution organisation and result persistence. *LLM model connectors* encapsulate the connection to a specific LLM and offer the function `generate_text`. With this function a benchmark task can send a prompt to LLM and get its answer. *Benchmark tasks* handle the LLM evaluation for a single task. In the function `evaluate_model` they usually

[2]Website: https://lm-kbc.github.io/challenge2023/
[3]Repository: https://github.com/google/BIG-bench
[4]Blogpost: https://lmsys.org/blog/2023-06-22-leaderboard/
[5]Repository: https://github.com/EleutherAI/lm-evaluation-harness

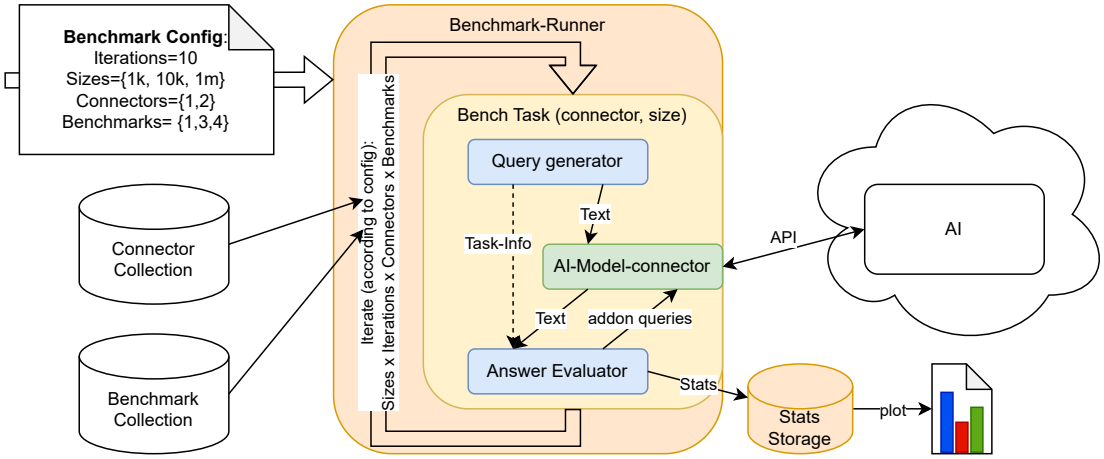


Figure 1: Basic *LLM-KG-Bench* framework architecture. The Benchmark runner takes a *benchmark configuration* and organizes the repeated execution of *benchmark tasks* with *LLM model connectors* and given size parameters. Results generated get stored and can be visualized.

Table 1
Setup used for testing the LLM-KG-Bench framework.

Model	Version
Claude	claude-1.3-100k
GPT 3.5	gpt-3.5-turbo-0613 (4k)
GPT 4	gpt-4-0613 (8k)

(a) LLMs evaluated

	Task a	Task b	Task c
Repetitions:	20 x 1 size	20 x 1 size	20 x 8 sizes
plot type:	F1 measure	F1 measure	Mean error
plot generated:	Figure 2a	Figure 2b	Figure 2c

(b) test configuration per task

build a prompt or task description for the LLM, hand this task over to a given LLM via an *LLM model connector* and evaluate the given answer. If necessary the *benchmark task* could send additional prompts to the LLM in the evaluation process. The evaluation results in score values for the task specific defined score types and additional information.

Due to *LLM-KG-Bench*'s modularization, as shown in Figure 1, additional benchmark tasks and LLM model connectors can be added by just adding corresponding python class definitions. The framework supports basic result visualization with the help of *seaborn*[6]. The plots shown in Figure 2 are generated this way.

4. Initial Evaluation of the Framework with first Tasks

To test the *LLM-KG-Bench* framework we added a couple of benchmark tasks and evaluated three of the currently highest ranking LLMs at the LLMSYS Chatbot Arena Leaderboard[4]. The test setup is detailed in Table 1.

[6]Website: https://seaborn.pydata.org/

Task a: Fixing of Errors in Turtle Files: Turtle is a common serialization format for knowledge graphs. By asking the LLMs to fix errors in given manipulated turtle files we test the knowledge of turtle syntax as well as strict adhering to the given task and facts. One of the scores calculated during evaluation is the F1 measure on parsable normalized triples, comparing LLM's answer with a perfect answer. A plot on the F1 measure results for this task is shown in Figure 2a. GPT-3.5 often claims that file would be correct and returns no turtle. This accounts for the high frequency of zero-value F1 scores. The answers given by Claude-1.3 and GPT-4 score better.

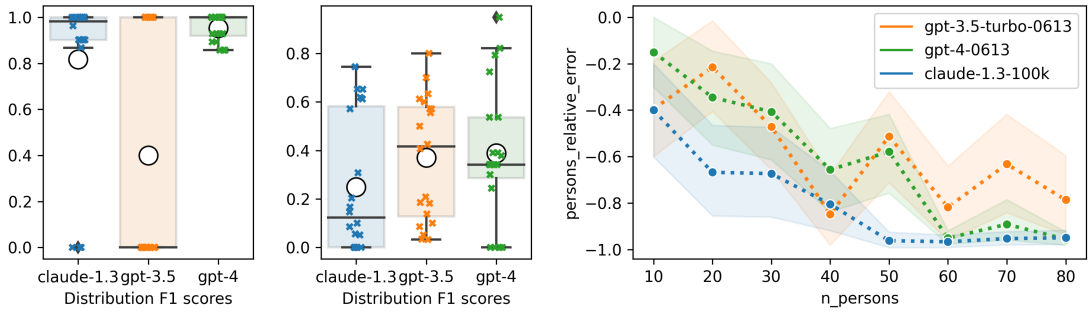
Task b: KG Creation from Factsheet Plaintext: To evaluate knowledge extraction and modelling capabilities, we use a plaintext excerpt of a PDF factsheet. The text describes various specifications of a 3D printer in a key-value style, including usual formatting irregularities associated with PDF extraction. We ask the model to generate a Turtle file, that captures a subset of the information. The prompt is engineered very specific with regard to which properties or ontologies have to be used and how IRI identifiers and Literals should be represented. Subsequently, we can evaluate the quality of a single response using the F1 measure, counting the set of parsable triples that (mis)match or are missing compared to a manually curated reference document. Fig. 2b shows that the GPT models outperform Claude in this task. While GPT4 has a better mean, due to one very good response, it however replied often with unparseable content, which in turn did not happen for GPT3.5, leading to a slightly better median for that.

Task c: Synthetic Dataset Generation: Creating example data is an important task and the help of LLMs would be highly appreciated. We created a basic test for this capability. We ask the LLM to generate some synthetic dataset using well known `foaf:Person` and `foaf:knows` with a varying number of desired objects and links in the final KG. In the evaluation we used beside other scores the *persons_relative_error* indicating the difference between the actual number person objects generated and the number asked for. This value is normalized to be $= 0$ if they match, > 0 if there are more persons than asked for and < 0 if there are less persons, with the special case of -1 meaning an empty graph. The results presented in Figure 2c show a relation between the *persons_relative_error* and the problem size, in this case number of person objects to generate.

5. Conclusion and Future Work

We showed that there is a need for measuring the knowledge graph engineering capabilities of the rapidly evolving LLMs. We proposed and describe the novel *LLM-KG-Bench* framework for this task. A first evaluation of three high ranking LLMs with first benchmarks shows the benefit of the automated evaluation with the new framework.

The *LLM-KG-Bench* framework is prepared to enable dialogs between benchmark tasks and LLMs. It will be interesting to evaluate LLMs capabilities to fix their answers with some feedback like e.g. error codes in improved or additional tasks. We are looking forward to extending to more LLMs and more benchmark tasks with the help of a bigger community.



(a) Turtle Fixing (b) Fact Extraction (c) Mean Error Dataset Generation

Figure 2: Subset of metrics from initial tasks. Shown are the F1 scores and mean error of person count

Acknowledgments

This work was partially supported by grants from the German Federal Ministry for Economic Affairs and Climate Action (BMWK) to the CoyPu project (01MK21007A) and KISS project (01MK22001A) as well as from the German Federal Ministry of Education and Research (BMBF) to the projects StahlDigital (13XP5116B) and KupferDigital (F13XP5119F).

References

[1] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[2] A. Srivastava, et al., Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, Transactions on Machine Learning Research (2023). arXiv:2206.04615.

[3] P. Groth, E. Simperl, M. van Erp, D. Vrandečić, Knowledge graphs and their role in the knowledge engineering of the 21st century (dagstuhl seminar 22372) (2023). doi:10.4230/DAGREP.12.9.60.

[4] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, 2023. arXiv:2306.08302.

[5] L.-P. Meyer, C. Stadler, J. Frey, N. Radtke, K. Junghanns, R. Meissner, G. Dziwis, K. Bulert, M. Martin, Llm-assisted knowledge graph engineering: Experiments with chatgpt, 2023. arXiv:2307.06917, to appear in proceedings of AI-Tomorrow track on Data Week 2023 in Leipzig.

A. Online Resources

- *LLM-KG-Bench* repository: https://github.com/AKSW/LLM-KG-Bench or doi:10.5281/zenodo.8251944
- experiment data: https://github.com/AKSW/LLM-KG-Bench-Results/tree/main/2023-SEMANTICS_LLM-KGE-Bench-Results or doi:10.5281/zenodo.8250646