Ontologies, Knowledge Graphs, and LLMs: How Do We GET Evaluations Done Right?

Heiko Paulheim[1]

[1]University of Mannheim, Germany

Abstract

The use of Large Language Models (LLMs) becomes increasingly popular for many tasks in the semantic web and knowledge graph community, e.g., knowledge graph (KG) construction, ontology learning, and ontology matching. Methods and tools using LLMs for those tasks are often evaluated on existing KGs and ontologies, which are publicly available on the Web. Thus, it is a reasonable assumption that the test data may have been seen by the LLM, and it is questionable if the results transfer to a case of unseen data (which is where those models are intended to be employed).

In this paper, we question the current evaluation paradigm using public data and propose a different approach, i.e., using a secondary LLM to create ontologies and knowledge graphs for one-time use on the fly. We coin this approach GET (generate–evaluate–trash). This also allows for repeating experiments and computing standard deviations and confidence intervals, which facilitates additional statements about the robustness of different approaches. We demonstrate our suggested approach on the case of taxonomy induction.

Keywords

Large Language Models, Ontologies, Knowledge Graphs, Data Leakage, Taxonomy Induction

1. Introduction

Large Language Models (LLMs) have become increasingly popular for many tasks in the semantic web and knowledge graph field [1, 2, 3], including ontology construction [4, 5], ontology refinement and validation [6, 7, 8], knowledge graph population [9], and ontology matching [10]. They are very promising both due to their straight forward usage, as well as the amount of knowledge they have ingested from large corpora during pre-training.

Evaluations of such approaches are often conducted on popular, publicly available ontologies and knowledge graphs, such as WordNet, Wikidata, the Gene Ontology, etc. This leads to a considerable problem in the significance of those evaluations: it is likely that the LLM has seen the evaluation data during training, a problem known as *data leakage*. While this problem is known in principle [11, 12, 13], there are only few proposals for solutions. Most of them address the challenge of *detecting* data leakage, but proposals for alternative evaluation protocols are still scarce. Moreover, the problem is particularly prominent in the semantic web and knowledge graphs community, where sharing ontologies and knowledge graphs as public artifacts is an explicit desideratum. With newer LLM-based AIs being increasingly equipped with the capability of using live Web search results for providing answers, and/or learning and based on user-input[1], evaluations on public benchmarks makes the evaluation results less and less significant and leads to a *vicious circle of AI evaluation*, as shown in Fig. 1.

This observation may be critical for applying LLM-based solutions in real-world scenarios, where the target data is not known, and where the good results on public benchmarks may lead to expectations which cannot met in practice. Consequently, recent works have already questioned the transferability to truly unseen domains, and shown that evaluation results obtained on public datasets are overly optimistic [14].

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan
heiko.paulheim@uni-mannheim.de (H. Paulheim)
http://www.heikopaulheim.com/ (H. Paulheim)
0000-0003-4386-8195 (H. Paulheim)

[1]https://www.linkedin.com/feed/update/urn:li:activity:7370135518602493952/

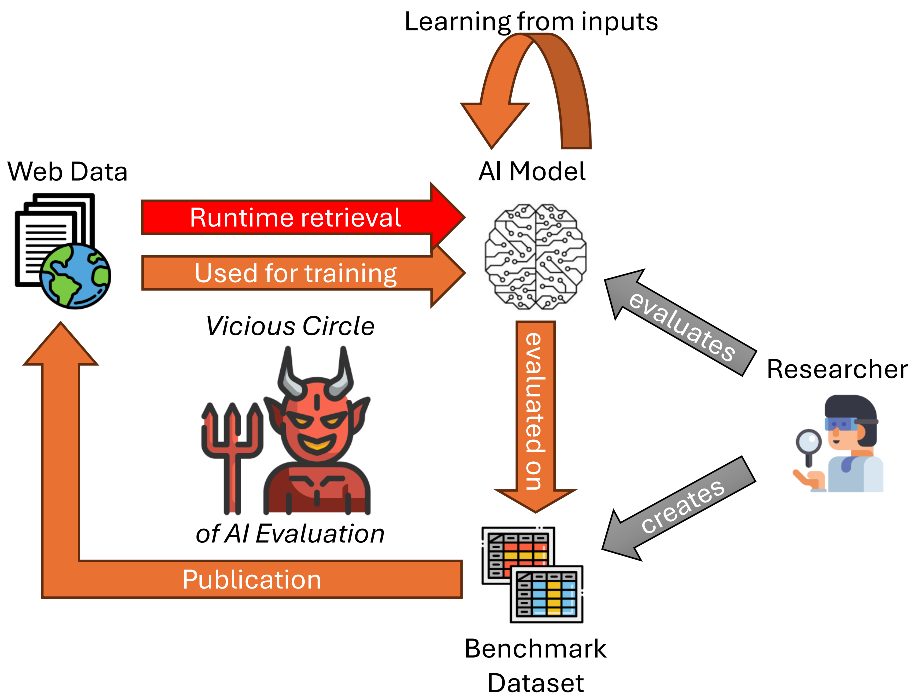

Figure 1: The vicious circle of AI evaluation. There are multiple ways in which benchmarks can become known to an AI model: by being part of the training data, by being retrieved at run-time, or by learning from user inputs when evaluating based on a benchmark.

2. Proposed Approach

In order to overcome the data leakage problem in evaluation, we propose a schema as shown in figure 2. We coin this the *GET methodology* (generate–evaluate–trash). It foresees the usage of a large language model to generate synthetic ontologies for one-time usage. In detail, the pipeline has the following steps:

1. From an original ontology, we extract key characteristics, such as the number of classes and properties.
2. The extracted characteristics are used to prompt an LLM to generate a set of synthetic ontologies resembling the original one. We propose two variants: (a) generating ontologies in the same domain, and (b) generating ontologies in related domains.
3. The result is a set of generated synthetic ontologies which have not been seen by any LLM during pre-training.
4. The synthetic ontologies are used as benchmarks for testing LLM-based tools, e.g., for ontology learning.
5. The results are collected. Since multiple similar ontologies can be generated, the approach also allows for assessing the stability of the results in addition to metrics such as precision and recall (e.g., by computing standard deviations across all generated ontologies).
6. After running the experiments, the synthetic ontologies should **not** be reused, but they can be made public in a research data repository for fostering reproducibility.

In step 2, in order to generate different ontologies, we propose using a temperature above 0. Moreover, we propose to use an LLM in this step which is not by any tool used in step 4.

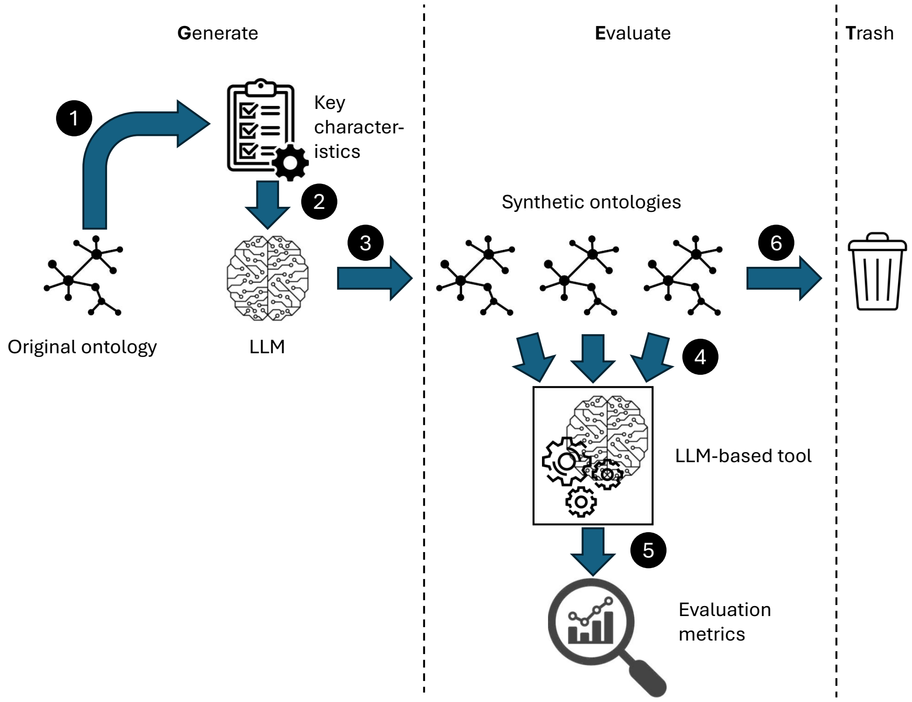

Figure 2: The GET methodology for running evaluations with LLM-based tools for ontologies

3. Example: Taxonomy Induction with LLMs

To test the proposed approach, we run experiments with taxonomy induction on two well-known ontologies, the Pizza ontology[2] and the Wine ontology[3]. For each of those, we asked an LLM to create three replica within the same domain, and three in adjacent domains (pasta, sushi, and curry dishes for the pizza ontology, and beer, whiskey, and gin for the wine domain). Details can be found in [15].

For each of those ontologies, we provide a list of all classes to an LLM, and ask it to return the subclass axioms holding between those classes. The returned subclass axioms are then compared to the one in the original ontology to compute recall, precision, and f-measure. The prompts used for generating the synthetic ontologies and for learning subclass axioms, as well as the generated ontologies, are available online.[4]

In our experiment, we use three LLMs of different sizes for taxonomy induction, i.e., Llama 8B, Llama 70B, and Mistral Large (123B) at a temperature of 0. The ontologies themselves are generated using Gemma-27B at a temperature of 0.5 (in order to create different test ontologies). The results are shown in table 1. We can make multiple observations:

1. The results on the original ontologies are often worse than those on the generated ones. There are at least two possible explanations: (a) the "mental models" of the generating and the evaluation LLMs are more aligned (i.e., LLMs, even different ones, have a certain shared understanding of a given domain), and (b) the original ontologies were created for instructive purposes, with the goal of displaying more different OWL constructs rather than providing a complete domain ontology.[5]
2. The results in related domains are generally worse than those in the original domain, especially in the tasks based on the wine ontology (i.e., beer, gin, and whiskey ontologies). This may hint at the LLMs having gathered a part of their ontology engineering knowledge on the wine ontology and related tutorial materials.

[2]https://protege.stanford.edu/ontologies/pizza/pizza.owl
[3]https://www.w3.org/TR/owl-guide/wine.rdf
[4]https://github.com/HeikoPaulheim/llm-ontology-learning
[5]For example, the generated pizza ontologies, on average, contain three times more different types of pizza than the original pizza ontology.

	pizza original			pizza'			pizza"			pizza"'			avg.		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B	0,286	0,253	0,268	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000 ± 0,000	0,000 ± 0,000	0,000 ± 0,000
Llama 70B	0,762	0,753	0,757	0,840	0,609	0,706	0,990	0,917	0,952	0,286	0,778	0,418	0,705 ± 0,371	0,768 ± 0,154	0,692 ± 0,267
Mistral Large	0,560	0,635	0,595	0,860	0,381	0,528	0,990	0,980	0,985	0,806	0,699	0,749	0,885 ± 0,095	0,687 ± 0,300	0,754 ± 0,229
				pasta			curry			sushi			avg.		
				r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B				0,120	0,088	0,101	0,034	0,034	0,034	0,594	0,383	0,466	0,249 ± 0,302	0,168 ± 0,188	0,200 ± 0,232
Llama 70B				0,740	0,587	0,655	0,853	0,779	0,814	0,739	0,543	0,626	0,777 ± 0,065	0,636 ± 0,126	0,698 ± 0,101
Mistral Large				0,830	0,654	0,731	0,863	0,788	0,824	0,623	0,473	0,538	0,772 ± 0,130	0,638 ± 0,159	0,698 ± 0,146
	wine original			wine'			wine"			wine"'			avg.		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B	0,662	0,305	0,418	0,919	0,782	0,845	0,700	0,275	0,394	1,000	0,363	0,532	0,873 ± 0,155	0,473 ± 0,271	0,591 ± 0,231
Llama 70B	0,761	0,388	0,514	0,884	0,784	0,831	0,175	1,000	0,298	1,000	1,000	1,000	0,686 ± 0,447	0,928 ± 0,125	0,709 ± 0,366
Mistral Large	0,310	0,289	0,299	0,884	0,784	0,831	0,900	0,621	0,735	1,000	0,965	0,982	0,928 ± 0,063	0,790 ± 0,172	0,849 ± 0,125
				beer			whiskey			gin			avg.		
				r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B				0,890	0,702	0,785	0,000	0,000	0,000	0,355	0,196	0,253	0,415 ± 0,448	0,299 ± 0,362	0,346 ± 0,401
Llama 70B				0,476	0,342	0,398	0,523	0,288	0,371	0,952	0,584	0,724	0,650 ± 0,262	0,405 ± 0,158	0,498 ± 0,196
Mistral Large				0,476	0,307	0,373	0,614	0,482	0,540	0,919	0,576	0,708	0,670 ± 0,227	0,455 ± 0,136	0,540 ± 0,167

Table 1: Results for taxonomy induction. The averages and standard deviations are only computed on the generated ontologies, excluding the original ones.

3. The order of tools by performance is not the same. For example, while Llama70B is superior to Mistral Large on almost all tasks on the original ontologies, Mistral Large outperforms Llama70B on many of the generated ontologies (both in the same and in similar domains). This may hint at a higher tendency of Llama70B's results being an effect of memorization to a larger extent than Mistral Large. This change of ordering demonstrates that evaluating on synthetic ontologies can reveal additional information that the evaluation on original ontologies do not provide.
4. The standard deviation is often considerable, showing that the approaches are not very stable, that good results can also be the result of a lucky coincidence, and that results in the same quality cannot be guaranteed on unseen data.

Overall, the results demonstrate that with the GET methodology, we can obtain more in-depth results than by only evaluating on the two original ontologies.

4. Conclusion and Outlook

Test data leakage is an overlooked issue when running LLM-based tools and evaluating them on public ontologies and knowledge graphs. In this paper, we have proposed the GET (generate–evaluate–trash) methodology as an alternative: instead of evaluating against publicly available knowledge graphs and ontologies, we propose to generate those on the fly for one-time evaluations. We have demonstrated the approach on the task of taxonomy induction, showing that it is possible to evaluate and also assess robustness of LLM-based taxonomy induction mechanisms.

First and foremost, future work will consist of wrapping the approach in an end-to-end evaluation pipeline. Further experimentation will go into controlling the complexity and difficulty of the generated ontologies, and the conduction of experiments in other tasks than taxonomy induction.

Acknowledgments

The experiments have been run using the Chat AI service provided by GWDG. [16]

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] N. Fathallah, A. Das, S. D. Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, Neon-gpt: a large language model-powered pipeline for ontology learning, in: European Semantic Web Conference, Springer, 2024, pp. 36–50.

[2] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, IEEE Transactions on Knowledge and Data Engineering (TKDE) (2024).

[3] C. Shimizu, P. Hitzler, Accelerating knowledge graph and ontology engineering with large language models, Journal of Web Semantics 85 (2025) 100862.

[4] B. Chen, F. Yi, D. Varró, Prompting or fine-tuning? a comparative study of large language models for taxonomy construction, in: 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), IEEE, 2023, pp. 588–596.

[5] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An llm supported approach to ontology and knowledge graph construction, arXiv preprint arXiv:2403.08345 (2024).

[6] Y. Zhao, N. Vetter, K. Aryan, Using large language models for ontoclean-based ontology refinement, arXiv preprint arXiv:2403.15864 (2024).

[7] S. Tsaneva, S. Vasic, M. Sabou, Llm-driven ontology evaluation: Verifying ontology restrictions with chatgpt, The semantic web: ESWC satellite events 2024 (2024).

[8] D. Shu, T. Chen, M. Jin, C. Zhang, M. Du, Y. Zhang, Knowledge graph large language model (kg-llm) for link prediction, Proceedings of Machine Learning Research 260 (2024) 143–158.
[9] S. S. Norouzi, A. Barua, A. Christou, N. Gautam, A. Eells, P. Hitzler, C. Shimizu, Ontology population using llms, in: Handbook on Neurosymbolic AI and Knowledge Graphs, IOS Press, 2025, pp. 421–438.
[10] S. Hertling, H. Paulheim, Olala: Ontology matching with large language models, in: Proceedings of the 12th knowledge capture conference 2023, 2023, pp. 131–139.
[11] K. Zhou, Y. Zhu, Z. Chen, W. Chen, W. X. Zhao, X. Chen, Y. Lin, J.-R. Wen, J. Han, Don't make your llm an evaluation benchmark cheater, arXiv preprint arXiv:2311.01964 (2023).
[12] Y. Cheng, Y. Chang, Y. Wu, A survey on data contamination for large language models, arXiv preprint arXiv:2502.14425 (2025).
[13] S. Ni, X. Kong, C. Li, X. Hu, R. Xu, J. Zhu, M. Yang, Training on the benchmark is not all you need, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, 2025, pp. 24948–24956.
[14] H. T. Mai, C. X. Chu, H. Paulheim, Do llms really adapt to domains? an ontology learning perspective, in: International Semantic Web Conference, Springer, 2024, pp. 126–143.
[15] H. Paulheim, Towards evaluating knowledge graph construction and ontology learning with llms without test data leakage, in: 3rd workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM), 2025.
[16] A. Doosthosseini, J. Decker, H. Nolte, J. M. Kunkel, Chat ai: A seamless slurm-native solution for hpc-based services, 2024. URL: https://arxiv.org/abs/2407.00110. arXiv:2407.00110.