Towards Evaluating Knowledge Graph Construction and Ontology Learning with LLMs without Test Data Leakage

Heiko Paulheim[1]

[1]University of Mannheim, Germany

Abstract

The use of Large Language Models (LLMs) becomes increasingly popular for knowledge graph construction and ontology learning. Very often, methods and tools using LLMs for those tasks are evaluated on existing knowledge graphs and ontologies, which are publicly available on the Web. In cases of very popular ontologies and knowledge graphs, there might be additional material such as tutorials and publications. Thus, it can be assumed that the test data has 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 propose a different method of evaluating LLMs for knowledge graph construction and ontology learning. We suggest using a secondary LLM to create test data for one-time use on the fly. 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 two original ontologies, and discuss different observations when comparing results between original and generated test data.

Keywords

Large Language Models, Ontology Learning, Evaluation, Data Leakage

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 done 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. While this problem is known in principle [11, 12], 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 open data is an explicit desideratum.

This observation may be critical for applying LLM-based ontology learning in real scenarios, where the target data is not known, because it cannot be trivially assumed that approaches tested on public ontologies behave alike on unseen data. Consequently, recent works have already questioned the transferability of LLM-based ontology learning methods to truly unseen domains, and shown that evaluation results obtained on public datasets are overly optimistic [13].

To overcome those problems and facilitate the evaluation of LLM-based ontology learning tools without data leakage, we propose an approach which foresees the usage of a large language model to generate synthetic ontologies for one-time usage. To that end, we adapt the GET (generate, evaluate, thrash) methodology proposed in [14] for ontology learning.

Joint proceedings of KBC-LM and LM-KBC @ ISWC 2025

heiko.paulheim@uni-mannheim.de (H. Paulheim)

http://www.heikopaulheim.com/ (H. Paulheim)

0000-0003-4386-8195 (H. Paulheim)

2. Related Work

While a considerable amount of works has been conducted on using LLMs for ontology learning, the evaluation is most often conducted on well-known, public ontologies.

The LLMs4OL challenge, conducted for the first time at the International Semantic Web Conference in 2024, uses WordNet, GeoNames, UMLS, the Gene Ontology (GO), FoodOn, and schema.org as test ontologies. [15] The 2025 edition[1] added the Ontology for Biomedical Investigations (OBI), the Material Ontology (MatOnto), Semantic Web for Earth and Environment Technology Ontology (SWEET), the Human Disease Ontology (DOID), the PROcess Chemistry Ontology (PROCO), and the Plant Ontology (PO) – all of which are publicly available and widely used ontologies. Similarly, the OntoURL benchmark uses a larger set of publicly available ontologies, including many of the aforementioned ones. [16]

The KBC-LM challenge, conducted for the first time as a challenge at the International Semantic Web Conference, has been using relations from Wikidata throughout all iterations. [17] Evaluation datasets proposed in other papers use taxonomies such as those from arxiv.org and Wikipedia [18], or public ontologies such as DOREMUS, Polifonia, DemCar, Odeuropa, NORIA-O, or FIBO [19].

Another strain of works (e.g. [20, 21]) does not evaluate the generated ontologies against a ground truth, but rather uses quality metrics such as those defined by the ontology pitfalls scanner (OOPS!) [22]. While this avoids the data leakage problem, it can rate only the compliance of LLM-based approaches with ontology engineering guidelines and finds general issues such as taxonomy cycles, but does not take the actual semantics of the generated ontologies into account.

Overall, we see that there is no easy way to evaluate how well LLM-based approaches work for ontology learning on unseen data.

The approach in this paper proposes to use synthetic ontologies as benchmarks for ontology learning, which are created dynamically for an experiment, and not reused afterwards. While synthethic ontologies have been proposed for other benchmarking means, such as reasoning [23, 24], knowledge graph completion [25], machine learning over knowledge graphs [26], or querying [27, 28, 29], the approach pursued in this paper differs in the two aspects that such approaches do not exist for ontology learning, and that the generation at runtime has not been in the focus so far (in fact, most synthetic benchmarks are public, and usually, researchers reuse public synthetic benchmarks instead of recreating fresh ones).

3. Proposed Approach

In order to overcome the data leakage problem in evaluating LLM-based ontology learning tools, we propose a schema based on the GET methodology [14], as shown in Figure 1. 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 that are generated on the fly. We assume that they were not part of the LLM training data.
4. The synthetic ontologies are used as benchmarks for testing LLM-based ontology learning tools.
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.

[1] https://sites.google.com/view/llms4ol2025/home

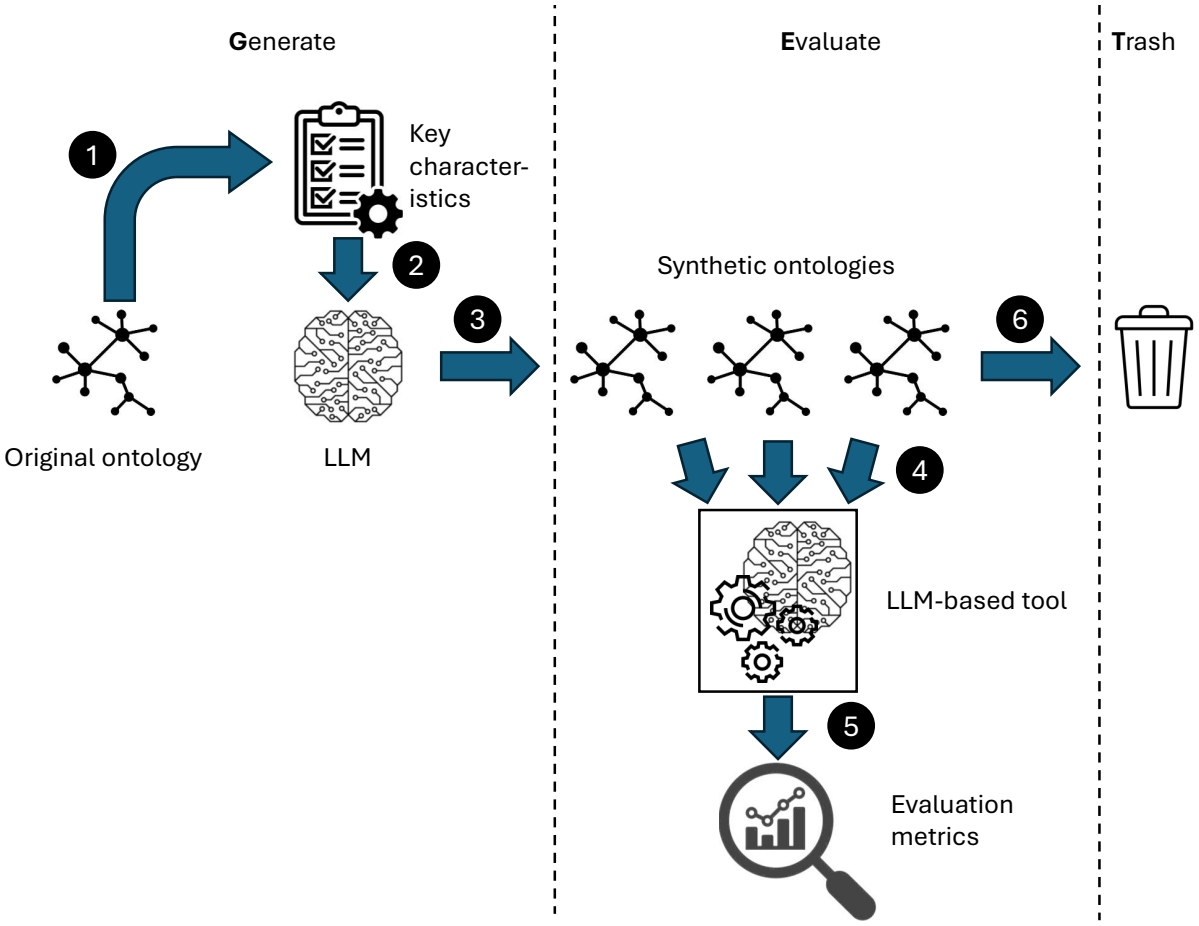

Figure 1: Proposed evaluation pipeline, adapted from [14], using the three phases *generate*, *evaluate*, and *trash*.

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

4. Experiments

In order to test the proposed approach, we conducted experiments with two ontology learning tasks, i.e., taxonomy induction and domain/range induction.

4.1. Ontology Generation

To test the proposed approach, we started from two well-known ontologies, the Pizza ontology[2] and the Wine ontology[3]. For each of those, we had an LLM create three replica within the same domain, and three in adjacent domains (pasta, sushi, and curry dishes for the pizza ontology, and beer, whiskey[4], and gin for the wine domain).[5] The prompts used for generating the synthetic ontologies, as well as for learning subclass and domain/range axioms, are shown in the appendix.

Statistics on the generated ontologies are shown in Table 1. We can make multiple observations here. First, while the LLM does a good job at creating an exact small number of items (here: properties), there is more variation for the larger numbers (here: classes). Second, while in the original ontologies, some properties do not have a defined domain or range, this never occurs in the generated ones, even though

[2]https://protege.stanford.edu/ontologies/pizza/pizza.owl
[3]https://www.w3.org/TR/owl-guide/wine.rdf
[4]Running the experiment with whisky and comparing the results to those with whiskey is left as an exercise to the reader.
[5]While we selected the adjacent domains by hand, it would also be possible to prompt an LLM for those for full automation.

	pizza original	pizza'	pizza"	pizza"'	pasta	sushi	curry
# classes	99	115	119	144	138	115	105
# properties	8	8	8	8	8	8	8
# subclass axioms	84	50	100	98	100	69	95
# domain axioms	6	8	9	8	8	8	8
# range axioms	7	8	8	8	8	8	8
	wine original	wine'	wine"	wine"'	beer	gin	whiskey
# classes	76	123	95	99	129	131	112
# properties	13	13	13	13	13	13	13
# subclass axioms	71	86	40	82	82	62	44
# domain axioms	6	13	13	13	13	13	13
# range axioms	9	13	13	13	13	13	13

Table 1
Characteristics of original and generated ontologies. The metrics derived from the original ontologies are used in prompts for the LLM to generate synthetic ones.

	Classes					Properties			
	wine	wine'	wine"	wine"'		wine	wine'	wine"	wine"'
wine	1	0.118	0.132	0.108	wine	1	0.000	0.083	0.040
wine'		1	0.298	0.291	wine'		1	0.130	0.130
wine"			1	0.252	wine"			1	0.238
wine"'				1	wine"'				1
	pizza	pizza'	pizza"	pizza"'		pizza	pizza'	pizza"	pizza"'
pizza	1	0.019	0.079	0.034	pizza	1	0.067	0.143	0.143
pizza'		1	0.109	0.122	pizza'		1	0.333	0.333
pizza"			1	0.297	pizza"			1	0.455
pizza"'				1	pizza"'				1

Table 2
Similarity of original and generated ontologies in terms of shared classes and properties. The table depicts the Jaccard overlap between of classes and properties.

this has been explicitly permitted in the prompt used to generate the ontologies. Moreover, in most cases, the domain of all properties is the central class (e.g., pizza or wine).

Table 2 shows the similarity of the original and generated ontologies in terms of overlapping classes and properties. It can be observed that the generated ontologies are very different from the original ontologies in that respect, and that the different generated ontologies are also reasonably different from one another.

4.2. Ontology Learning Evaluation

We evaluate two tasks, i.e., subclass axiom induction and domain/range induction by LLMs, and we use three LLMs of different sizes for that task: Llama 8B, Llama 70B, and Mistral Large Instruct (123B) at a temperature of 0 for reproducbility. The ontologies themselves are generated using Gemma-27B using a temperature of 0.5 to ensure variance in the generated ontologies.

Since the original ontologies were not fully materialized, we (a) materialized the domain/range axioms for inverse properties, and (b) added subclass axioms for equivalent restriction definitions (i.e., for $A \equiv B \sqcap C$, we added $A \sqsubseteq B$ and $A \sqsubseteq C$) before evaluating the generated domain/range and subclass axioms. Both sets of materialized axioms are included in the counts in Table 1. All generated ontologies and axioms output by the different models are available online.[6]

The results are shown in tables 3 for subclass induction, 4 for domain induction, and 5 for range induction.

Before analyzing the results in more detail, we want to point out two cases observed frequently throughout the entire evaluation:

[6] https://github.com/HeikoPaulheim/llm-ontology-learning

	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 3: Results for subclass induction (recall, precision, and F1-measure)

	pizza original			pizza’			pizza”			pizza”’			avg.		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B	0.625	0.006	0.012	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.063	0.119	0.333 ± 0.577	0.021 ± 0.037	0.040 ± 0.069
Llama 70B	0.500	0.500	0.500	0.875	0.875	0.875	0.990	0.917	0.952	1.000	0.667	0.800	0.955 ± 0.069	0.819 ± 0.134	0.876 ± 0.076
Mistral Large	0.500	0.500	0.500	1.000	1.000	1.000	0.990	0.980	0.985	1.000	1.000	1.000	0.997 ± 0.006	0.993 ± 0.011	0.995 ± 0.009
				pasta			curry			sushi			avg.		
				r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B				0.125	0.125	0.125	1.000	0.010	0.019	0.875	0.038	0.074	0.667 ± 0.473	0.058 ± 0.060	0.073 ± 0.053
Llama 70B				0.000	0.000	0.000	1.000	1.000	1.000	0.083	0.100	0.091	0.361 ± 0.555	0.367 ± 0.551	0.364 ± 0.553
Mistral Large				0.875	0.875	0.875	1.000	1.000	1.000	1.000	1.000	1.000	0.958 ± 0.072	0.958 ± 0.072	0.958 ± 0.072
	wine original			wine’			wine”			wine”’			avg.		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B	0.077	0.071	0.074	0.615	0.027	0.051	0.000	0.000	0.000	0.923	0.197	0.324	0.513 ± 0.470	0.074 ± 0.107	0.125 ± 0.174
Llama 70B	0.385	0.385	0.385	1.000	1.000	1.000	1.000	1.000	1.000	0.923	0.923	0.923	0.974 ± 0.044	0.974 ± 0.044	0.974 ± 0.044
Mistral Large	0.308	0.308	0.308	1.000	1.000	1.000	0.923	0.923	0.923	1.000	1.000	1.000	0.974 ± 0.044	0.974 ± 0.044	0.974 ± 0.044
				beer			whiskey			gin			avg.		
				r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B				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				1.000	1.000	1.000	0.769	0.769	0.769	0.923	0.923	0.923	0.897 ± 0.118	0.897 ± 0.118	0.897 ± 0.118
Mistral Large				0.000	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	0.667 ± 0.577	0.667 ± 0.577	0.667 ± 0.577

Table 4: Results for domain induction (recall, precision, and F1-measure)

	pizza original			pizza’			pizza”			pizza”’			avg.		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B	0.375	0.375	0.375	0.000	0.000	0.000	1.000	1.000	1.000	1.000	0.070	0.130	0.667 ± 0.577	0.357 ± 0.558	0.377 ± 0.544
Llama 70B	0.625	0.625	0.625	0.125	0.125	0.125	1.000	1.000	1.000	1.000	1.000	1.000	0.708 ± 0.505	0.708 ± 0.505	0.708 ± 0.505
Mistral Large	0.500	0.800	0.615	0.103	0.123	0.112	1.000	1.000	1.000	1.000	1.000	1.000	0.701 ± 0.518	0.708 ± 0.506	0.704 ± 0.513
				pasta			curry			sushi			avg.		
				r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B				0.875	0.778	0.824	0.625	0.625	0.625	0.000	0.000	0.000	0.500 ± 0.451	0.468 ± 0.412	0.483 ± 0.430
Llama 70B				0.875	0.875	0.875	0.125	0.125	0.125	0.875	0.875	0.875	0.625 ± 0.433	0.625 ± 0.433	0.625 ± 0.433
Mistral Large				0.875	0.875	0.875	0.146	0.175	0.159	1.000	0.889	0.941	0.674 ± 0.461	0.646 ± 0.408	0.658 ± 0.434
	wine original			wine’			wine”			wine”’			avg.		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B	0.692	0.692	0.692	0.923	0.150	0.258	0.923	0.706	0.800	1.000	0.929	0.963	0.949 ± 0.044	0.595 ± 0.401	0.674 ± 0.369
Llama 70B	0.692	0.692	0.692	1.000	1.000	1.000	0.846	0.846	0.846	1.000	1.000	1.000	0.949 ± 0.089	0.949 ± 0.089	0.949 ± 0.089
Mistral Large	0.692	0.692	0.692	1.000	1.000	1.000	0.923	0.923	0.923	1.000	1.000	1.000	0.974 ± 0.044	0.974 ± 0.044	0.974 ± 0.044
				beer			whiskey			gin			avg.		
				r	p	f	r	p	f	r	p	f	r	p	f
Llama 8B				0.923	0.923	0.923	1.000	0.260	0.413	1.000	1.000	1.000	0.974 ± 0.044	0.728 ± 0.407	0.779 ± 0.319
Llama 70B				1.000	1.000	1.000	0.769	0.769	0.769	1.000	1.000	1.000	0.923 ± 0.133	0.923 ± 0.133	0.923 ± 0.133
Mistral Large				0.923	0.923	0.923	0.846	0.846	0.846	1.000	1.000	1.000	0.923 ± 0.077	0.923 ± 0.077	0.923 ± 0.077

Table 5: Results for range induction (recall, precision, and F1-measure)

- Results where both recall and precision are 0 are in most cases due to the LLM answering with a completely different format than the one request. A particular second cause can be observed in particular for Llama-8B, which often mixes up domain and range, and outputs property ranges instead of domains, which is why Llama8B often has zeros in the domain induction task.
- Results with a very high recall and very low precision: occasionally, the LLMs output cross products of properties and classes for domains or ranges, or redundantly include all subclasses of the actual domain/range class.[7]

When looking into the results in more detail, we can make number of further observations:

- The results on the original ontologies are often worse than those on the generated ones. There are at least three possible explanations: (a) the "mental models" of the generating and the evaluation LLMs are more aligned (i.e., LLMs have a certain shared understanding of a given domain), (b) the original ontologies, which were created for instructive purposes, contain more corner cases, and (c) in contrast to most generated ontologies, the original ones contain properties without explicit domain and range definitions, while the LLM almost always returns a definition for each property, despite explicitly prompted that this is optional, leading to a larger number of false positives on the original ontologies.
- 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.
- 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.
- 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, we see that with the proposed methodology, we can obtain more in-depth results than by only evaluating on the two original ontologies.

5. Conclusion and Outlook

Test data leakage is an overlooked issue when evaluating LLM-based tools for ontology learning on publicly available ontologies and knowledge graphs. In this paper, we have proposed an alternative methodology: instead of evaluating against publicly available ontologies, we propose to generate test ontologies 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.

While we assume that the generated ontologies are not seen by the LLMs during training, this assumption may be partially wrong – in case the generating LLM reproduces parts of an existing ontology, those may in fact have been seen by the LLM. One important task of future work is therefore applying data leakage metrics [30] to the generated data to assess the degree of freshness of the generated synthetic ontologies.

One of the striking observations of this work was that the results observed on synthetic ontologies are often better than those on the original, human-generated ones. This deserves a deeper analysis. One possible reason we postulated was that different LLMs have a stronger alignment on their "mental models" of a domain, an assumption that deserves further analysis, e.g., by swapping the LLMs used

[7] In future work, we will catch the latter issue programmatically, and filter out those correct, but redundant axioms.

for generation and evaluation, and comparing the generated ontologies to one another. Moreover, we will think of approaches to assess the difficulty of the ontology learning tasks on the original and the generated ontologies, respectively, and to experiment with different prompts for controlling the task difficulty.

On the practical side, future work will consist of wrapping the approach in an end-to-end evaluation pipeline. Further experimentation will go into the generation of ontologies, e.g., controlling the complexity and difficulty of the generated ontologies, and conducting experiments with different generation models.

So far, we have looked into taxonomy and domain/range induction, but the approach might be interesting for various other tasks, such as the learning of more complex restrictions, detection of property characteristics (transitivity, inverse properties, etc.), or entity typing.

Overall, we hope that this mode of evaluation will be used at least in addition to the currently dominant paradigm of evaluating using publicly available ontologies, in order to analyze tool performance in a setup free from test data leakage.

Declaration on Generative AI

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

Acknowledgements

The experiments have been run using the Chat AI service provided by GWDG. [31]

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] 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.
[14] H. Paulheim, Ontologies, Knowledge Graphs, and LLMs: How Do We GET Evaluations Done Right?, in: International Semantic Web Conference, Posters and Demos, 2025.
[15] H. B. Giglou, J. D'Souza, S. Auer, Llms4ol 2024 overview: The 1st large language models for ontology learning challenge, arXiv preprint arXiv:2409.10146 (2024).
[16] X. Zhang, H. Lai, Q. Meng, J. Bos, Ontourl: A benchmark for evaluating large language models on symbolic ontological understanding, reasoning and learning, arXiv preprint arXiv:2505.11031 (2025).
[17] J.-C. Kalo, T.-P. Nguyen, S. Razniewski, B. Zhang, Preface: Lm-kbc challenge 2024, in: 2nd Workshop on Knowledge Base Construction from Pre-Trained Language Models, CEUR. ws, 2024.
[18] A. Lo, A. Q. Jiang, W. Li, M. Jamnik, End-to-end ontology learning with large language models, Advances in Neural Information Processing Systems 37 (2024) 87184–87225.
[19] Y. Rebboud, P. Lisena, L. Tailhardat, R. Troncy, Benchmarking llm-based ontology conceptualization: A proposal, in: ISWC 2024, 23rd International Semantic Web Conference, 2024.
[20] M. A. Cappelli, G. Di Marzo Serugendo, Methodological exploration of ontology generation with a dedicated large language model, Electronics 14 (2025) 2863.
[21] A. S. Lippolis, M. J. Saeedizade, R. Keskisärkkä, S. Zuppiroli, M. Ceriani, A. Gangemi, E. Blomqvist, A. G. Nuzzolese, Ontology generation using large language models, in: European Semantic Web Conference, Springer, 2025, pp. 321–341.
[22] M. Poveda-Villalón, M. C. Suárez-Figueroa, A. Gómez-Pérez, Validating ontologies with oops!, in: International conference on knowledge engineering and knowledge management, Springer, 2012, pp. 267–281.
[23] M. Ebrahimi, M. K. Sarker, F. Bianchi, N. Xie, D. Doran, P. Hitzler, Reasoning over rdf knowledge bases using deep learning, arXiv preprint arXiv:1811.04132 (2018).
[24] A. Eberhart, M. Ebrahimi, L. Zhou, C. Shimizu, P. Hitzler, Completion reasoning emulation for the description logic el+, in: Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice, 2020.
[25] A. Melo, H. Paulheim, Synthesizing knowledge graphs for link and type prediction benchmarking, in: European Semantic Web Conference, Springer, 2017, pp. 136–151.
[26] J. Portisch, H. Paulheim, The dlcc node classification benchmark for analyzing knowledge graph embeddings, in: International semantic web conference, Springer, 2022, pp. 592–609.
[27] Y. Guo, Z. Pan, J. Heflin, Lubm: A benchmark for owl knowledge base systems, Journal of Web Semantics 3 (2005) 158–182.
[28] M. Schmidt, T. Hornung, G. Lausen, C. Pinkel, Spˆ 2bench: a sparql performance benchmark, in: 2009 IEEE 25th International Conference on Data Engineering, IEEE, 2009, pp. 222–233.
[29] C. Bizer, A. Schultz, The berlin sparql benchmark, in: Semantic Services, Interoperability and Web Applications: Emerging Concepts, IGI Global Scientific Publishing, 2011, pp. 81–103.
[30] 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.
[31] 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.

Appendix

This section documents the prompts used in the experiments. In all cases, we used the following system prompt:

```
You are an ontology engineer
```

For the generation of ontologies, we used Gemma-27B with a temperature of 0.5 and the following prompt:

```
Build me an ontology of {domain} with {N} Classes and {M} ObjectProperties,
↪  please. Provide the result as
1. List of Classes (one per line)
2. List of ObjectProperties (one per line)
3. List of subclass axioms in the format "A,B" (one per line, where A is a
↪  subclass of B)
4. List of domain axioms in the format P,D (one per line, where P is a
↪  property, D is its domain - leave empty if there is no suitable class)
5. List of range axioms in the format P,R (one per line, where P is a
↪  property, R is its range - leave empty if there is no suitable class)
```

Here, `domain` is one out of `{pizza,pasta,sushi,curry dishes,wine,beer,whiskey,gin}`, and `N` and `M` are numbers extracted from the original pizza and wine ontologies. The output format is chosen to ease the evaluation.

For the generation of subclass axioms, the following prompt is used:

```
I want to build an ontology of {domain}. I give you the classes I defined,
↪  please provide a list of subclass axioms between those classes in the
↪  Format
A,B
where A rdfs:subClassOf B holds
List of Classes:
{classes}
```

For the generation of domain axioms, the following prompt is used:

```
I want to build an ontology of {domain}. I give you the classes and
↪  properties I defined, please provide a list of domain definitions in the
↪  format
P,D
where P rdfs:domain D holds.
Leave the fields empty if none of the classes defined is suitable as a
↪  domain.
List of Properties:
{properties}
List of Classes:
{classes}
```

For the generation of range axioms, the following prompt is used:

```
I want to build an ontology of {domain}. I give you the classes and
↪  properties I defined, please provide a list of range definitions in the
↪  format
P,R
where P rdfs:range R holds.
Leave the fields empty if none of the classes defined is suitable as a range.
List of Properties:
{properties}
List of Classes:
{classes}
```

In the latter three prompts, `{classes}` and `{properties}` are lists of the classes and properties of the ontology at hand, provided one per line.