LLMs4OL 2024: The 1st Large Language Models for Ontology Learning Challenge at the 23rd ISWC

LLMs4OL 2024 Task Participant Papers

https://doi.org/10.52825/ocp.v4i.2492

Published: 02 Oct. 2024

DSTI at LLMs4OL 2024 Task A: Intrinsic Versus Extrinsic Knowledge for Type Classification

Applications on WordNet and GeoNames Datasets

Hanna Abi Akl[1,2]

[1]Data ScienceTech Institute (DSTI), 4 Rue de la Collégiale, 75005, Paris, France

[2]Université Côte d'Azur, Inria, CNRS, I3S

*Correspondence: Hanna Abi Akl, hanna.abi-akl@dsti.institute

Abstract: We introduce semantic towers, an extrinsic knowledge representation method, and compare it to intrinsic knowledge in large language models for ontology learning. Our experiments show a trade-off between performance and semantic grounding for extrinsic knowledge compared to a fine-tuned model's intrinsic knowledge. We report our findings on the Large Language Models for Ontology Learning (LLMs4OL) 2024 challenge.

Keywords: Large Language Models, Ontology Learning, Semantic Web, Knowledge Representation, Semantic Primes

1 Introduction and related work

Large language models (LLMs) have seen widespread applications across different tasks in the fields of Natural Language Processing and Knowledge Representation. Particularly, LLM-based systems are used to tackle ontology-related tasks such as ontology learning [1], knowledge graph construction [2], ontology matching [3][4] and ontology generation [5]. Retrieval-Augmented-Generation (RAG) systems, which build on the capabilities of LLMs by enhancing retrieval using external knowledge sources, have also shown promising results in tasks involving the use of ontologies [6]. On the other hand, symbolic methods like semantic representation using primes and universals [7] form another research frontier in the area of knowledge representation which is at the heart of ontologies [8].

In this work, we evaluate and compare the performance of fine-tuned models on Task A of the LLMs4OL [9][10][11] 2024 challenge[1] using intrinsic LLM knowledge and external knowledge sources we define as semantic towers. The rest of the work is organized as follows. In section 2, we present our methodology. Section 3 describes our experimental framework. In section 4, we report our results and discuss our findings. Finally, we conclude in section 5.

[1]https://sites.google.com/view/llms4ol/home

2 Methodology

This section describes the methodology for creating a semantic tower *ST* which we define as:

$$ST = \{s_1, s_2, .., s_n\}, \tag{1}$$

where *s* is a domain semantic primitive pointing to a semantic property for a given domain and *n* is the minimal number of primitives needed to define the domain. The rest of this section details the construction of domain semantic towers from semantic primitives.

2.1 Domain semantic primitives

For each domain, we use the Wikidata Query Service[2] to retrieve semantic information for each term type category. This body of information, or semantic set, serves as the base for the domain semantic primitives.

The WordNet semantic set consists of: {subclass,instance,part,represents,description}. The GeoNames semantic set consists of: {subclass,instance,part,category,description}.

2.2 Semantic towers

The construction scheme of semantic towers is domain-invariant and summarized in the following steps:

1. The values of the semantic set for each term type are tokenized into a bag of words, cleaned and normalized through lowercase transformation and stop word removal.
2. The result is transformed to a comma-separated list.
3. Empty values and duplicates are pruned from the list.
4. The list of primitives is transformed to vector embeddings of size 1024 using the gte-large[3] model by Google [12].
5. The resulting domain vector embeddings are stored in a MongoDB[4] collection to form a vector store, i.e. the semantic tower.
6. The semantic tower is indexed on embeddings search for optimized performance.

Figure 1 shows examples of the WordNet and GeoNames semantic towers.

3 Experiments

This section describes our experiments in terms of data, models and training process.

3.1 Dataset description

We consider two datasets for our experiments: WordNet and GeoNames. Both datasets are used for training and testing our models in the respective subtasks (A.1 and A.2). The dataset descriptions are detailed in the following subsections.

[2] https://query.wikidata.org/
[3] https://huggingface.co/thenlper/gte-large
[4] https://www.mongodb.com/

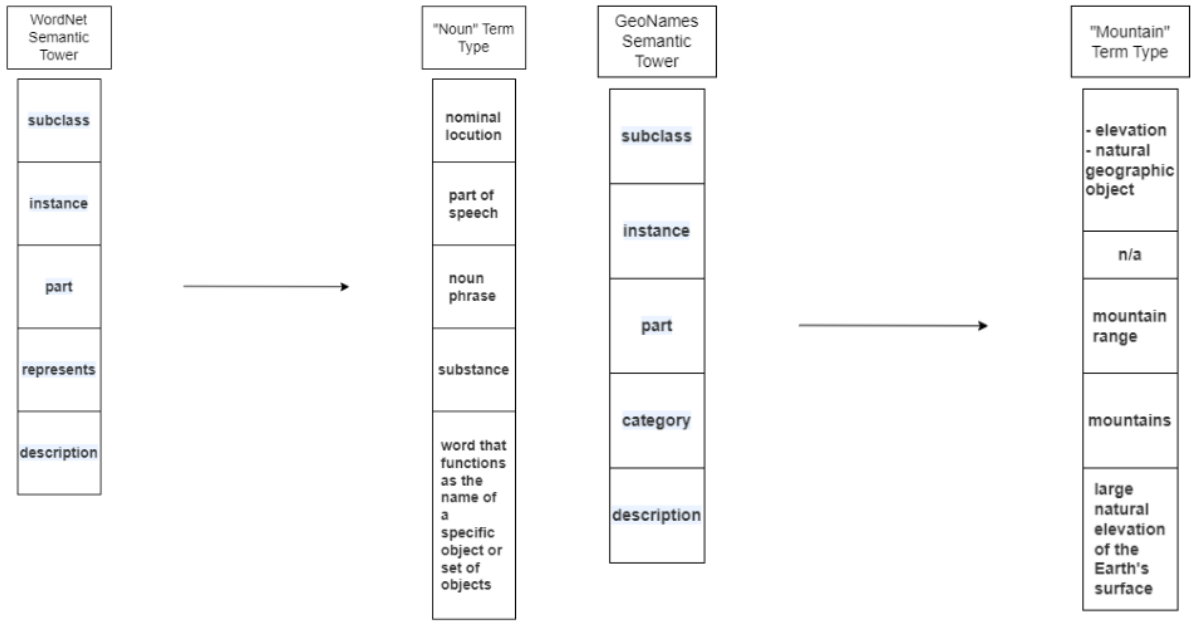

Figure 1. *WordNet and GeoNames semantic towers with examples.*

3.1.1 WordNet

The dataset consists of 40,559 train terms and 9,470 test terms. It contains four types to classify each term: noun, verb, adjective, adverb. Figure 2 shows example data.

Lexical Term L	Sentence Containing L (Optional)	Type
question	there was a question about my training	noun
lodge	Where are you lodging in Paris?	verb
genus equisetum		noun

Figure 2. *Subtask A.1 term typing WordNet examples.*

3.1.2 GeoNames

The dataset consists of 8,078,865 train terms and 702,510 test terms. It contains 660 categories of geographical locations. Example data is presented in Figure 3.

Lexical Term L	Type
Pic de Font Blanca	peak
Roc Mele	mountain
Estany de les Abelletes	lake

Figure 3. *Subtask A.2 term typing GeoNames examples.*

3.2 System description

This section describes the models as well as the setup of our experiments.

3.2.1 Models

We train one model for each subtask. We use the same base flan-t5-small[5] model and fine-tune it on the subtask datasets respectively. The training hyperparameters for both models are configured identically: {learning_rate: 1e-05, train_batch_size: 4, eval_batch_size: 4, num_epochs: 5, question_length: 512, target_length: 512, optimizer: Adam}. For subtask A.1, the model is trained on 70% of the provided WordNet dataset and the remaining 30% is used for validation. Table 1 shows the training results.

Table 1. *Subtask A.1 model training results.*

Training Loss	Epoch	Step	Validation Loss
0.1725	1.0	1000	0.0640
0.1250	2.0	2000	0.0535
0.1040	3.0	3000	0.0469
0.0917	4.0	4000	0.0421
0.0830	5.0	5000	0.0384

For subtask A.2, the length of the data makes fine-tuning challenging. To remedy this problem, we curate a subset from the original dataset using the following algorithm:

1. Each type category is counted into a length variable *cat_len*.
2. For each category represented less than 100 times (i.e. $cat\_len < 100$), all terms classified in that category are selected and kept in the dataset.
3. If $cat\_len \geq 100$, only the first 25 terms classified in that category are selected. The threshold of 25 keeps the size of the dataset relatively small given the large number of categories.

We obtain a curated dataset of 2041 terms representing all possible categories. The model is trained on 70% of the curated dataset and the remaining 30% is used for validation. Table 2 shows the training results.

Table 2. *Subtask A.2 model training results.*

Training Loss	Epoch	Step	Validation Loss
2.6223	1.0	1000	1.5223
2.1430	2.0	2000	1.3764
1.9100	3.0	3000	1.2825
1.7642	4.0	4000	1.2102
1.6607	5.0	5000	1.1488

The training of both models is done on a Google Colab instance using an A100 High-RAM GPU. Both A.1 and A.2 models are available publicly on Hugging Face respectively under the names flan-t5-small-wordnet[6] and flan-t5-small-geonames[7].

3.2.2 Features

The same feature engineering method is applied for both models. It consists in embedding input text into vectors of size 1024 using the gte-large model. For the flan-t5-small-wordnet model, the input is the concatenation of the term and the sentence when provided. For flan-t5-small-geonames, the input text is the term.

[5] https://huggingface.co/google/flan-t5-small
[6] https://huggingface.co/HannaAbiAkl/flan-t5-small-wordnet
[7] https://huggingface.co/HannaAbiAkl/flan-t5-small-geonames

3.2.3 Setup

We conduct two experiments per subtask for a total of four.

For subtask A.1, the first experiment (WN1) consists in prompting the fine-tuned WordNet model on the test split of the provided dataset which is used as an unofficial test set ahead of the official submission. The prompt used for the model is: ***Give the entity for the term X. Select the answer from this list Y***, where X is dynamically replaced by the input term and Y is replaced by the list of possible term types.

The second experiment (WN2) leverages the RAG pipeline shown in Figure 4 in conjunction with a user prompt to retrieve the best term type for each input term. The input is vectorized and compared to the embeddings of the WordNet semantic tower for each term type. A cosine similarity score is used to determine the closest type from the semantic tower vector store to return the top 1 candidate. The answer is then used as an additional input to the user prompt given to the model: ***Give the entity for the term X. Select the answer from this list Y relying on the search result Z***, where X and Y are as previously defined and Z represents the best-matched term type from the semantic tower.

For subtask A.2, both experiments GN1 and GN2 mimic WN1 and WN2 respectively. For GN1, the fine-tuned GeoNames model is evaluated on the test split of the curated dataset. The user prompt for the model is the same as that of WN1, with the only changes being the X term values and the Y list of types which now refers to the geographical categories.

In experiment GN2, the same pipeline from Figure 4 is reproduced with the only difference being the replacement of the WordNet semantic tower with the GeoNames semantic tower. The user prompt used for the fine-tuned model is the same as that of WN2, with the Y list reflecting the geographical categories. All experiments are conducted on a Google Colab instance using a L4 High-RAM GPU. The code for our experimental setup is publicly available on GitHub[8].

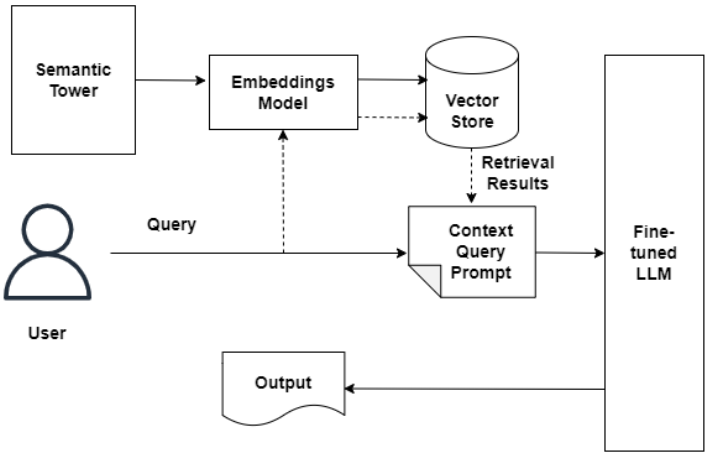

Figure 4. *RAG system architecture.*

[8] https://github.com/HannaAbiAkl/SemanticTowers

4 Results

Table 3 shows our experimental results on the WordNet test set. The results of the GeoNames experiments are presented in Table 4. The F1 scoring metric reflects the criteria of performance assessment set by the task organizers.

Experiments WN1 and GN1 perform better than WN2 and GN2 respectively, with a performance gain close to 10%. At first inspection, the results seem to suggest that the flan-t5 model, with a little fine-tuning, can rely on its existing knowledge regarding the dataset domains to correctly classify terms by type. The use of an external knowledge base, such as a semantic tower, seems to create more errors in the model answers. However, closer examination of a subset of the outputs reveals that semantic towers effectively ground certain semantic notions in the model that are otherwise lost if the model only relies on its existing knowledge. Examples include correctly classifying the term *into the bargain* as *adverb* with the aid of the WordNet semantic tower (as opposed to classifying it as *noun* without it). While the word *bargain* dominates the term in the example, the flan-t5-small-wordnet model misses out on the correct classification which attributes an important weight to the adverb *into* that becomes more prominent with the semantic tower embeddings representation. A similar case can be made for the GeoNames experiments, where the usage of the semantic tower in conjunction with the model improves the classification choice for plural categories (e.g. terms classified as *mountains*, *peaks*, *streams*). The outputs of experiment GN1 show that the model alone has a tendency to choose the singular forms of these categories which count for incorrect classifications. Moreover, experiment GN2 also shows that the semantic tower helps ground nuances between categories (e.g. *stream* versus *section of stream*) which leads to a more fine-grained (and accurate) typing.

For the official test sets released by the task organizers, we evaluate only the A.1 subtask using WN1 and WN2 and present our results in Table 5. Both WN1 and WN2 demonstrate a slight drop in performance of around 1% but perform competitively well. The results demonstrate that the model training as well as the WordNet semantic tower construction are sound enough to avoid catastrophic drift.

We refrain from submitting to the other subtasks, most notably A.2, because of the length of the official test set which is extremely challenging to run on our available resources.

Table 3. *Experimental results on the WordNet set.*

Experiment	F1
flan-t5-small-wordnet (WN1)	**0.9820**
flan-t5-small-wordnet + WordNet semantic tower (WN2)	0.8581

Table 4. *Experimental results on the GeoNames set.*

Experiment	F1
flan-t5-small-geonames (GN1)	**0.6820**
flan-t5-small-geonames + GeoNames semantic tower (GN2)	0.5636

Table 5. *Subtask A.1 (few-shot) WordNet term typing leaderboard.*

Teal Name	F1	Precision	Recall
TSOTSALearning	0.9938	0.9938	0.9938
DSTI (WN1)	**0.9716**	**0.9716**	**0.9716**
DaseLab	0.9697	0.9689	0.9704
RWTH-DBIS	0.9446	0.9446	0.9446
TheGhost	0.9392	0.9389	0.9395
Silp_nlp	0.9037	0.9037	0.9037
DSTI (WN2)	**0.8420**	**0.8420**	**0.8420**
Phoenixes	0.8158	0.7689	0.8687

5 Conclusion

In this shared task, we investigate and compare intrinsic knowledge in LLMs with external semantic sources for ontology learning. While the introduction of semantic towers proves there is still some way to go to achieve semantic resonance in LLMs, it shows promising results in grounding these models semantically and fine-graining their knowledge. Our fine-tuned models demonstrate that ontology term typing is a task within the reach of LLMs based on their existing knowledge. In future work, we will explore the potential of semantic towers and expand their implementation to existing LLM-based systems.

Author Contributions

Hanna Abi Akl: The author solely contributed to the work.

Competing interests

The authors declare that they have no competing interests.

References

[1] F. Ronzano and J. Nanavati, "Towards ontology-enhanced representation learning for large language models," *arXiv preprint arXiv:2405.20527*, 2024.

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

[3] H. B. Giglou, J. D'Souza, and S. Auer, "Llms4om: Matching ontologies with large language models," *arXiv preprint arXiv:2404.10317*, 2024.

[4] Y. He, J. Chen, H. Dong, and I. Horrocks, "Exploring large language models for ontology alignment," *arXiv preprint arXiv:2309.07172*, 2023.

[5] S. Toro, A. V. Anagnostopoulos, S. Bello, *et al.*, "Dynamic retrieval augmented generation of ontologies using artificial intelligence (dragon-ai)," *arXiv preprint arXiv:2312.10904*, 2023.

[6] M. J. Buehler, "Generative retrieval-augmented ontologic graph and multiagent strategies for interpretive large language model-based materials design," *ACS Engineering Au*, vol. 4, no. 2, pp. 241–277, 2024.

[7] A. Wierzbicka, *Semantics: Primes and universals: Primes and universals*. Oxford University Press, UK, 1996.

[8] J. Fähndrich, *Semantic decomposition and marker passing in an artificial representation of meaning*. Technische Universitaet Berlin (Germany), 2018.

[9] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," in *International Semantic Web Conference*, Springer, 2023, pp. 408–427.

[10] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol 2024 overview: The 1st large language models for ontology learning challenge," *Open Conference Proceedings*, vol. 4, Oct. 2024.

[11] H. Babaei Giglou, J. D'Souza, S. Sadruddin, and S. Auer, "Llms4ol 2024 datasets: Toward ontology learning with large language models," *Open Conference Proceedings*, vol. 4, Oct. 2024.

[12] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, "Towards general text embeddings with multi-stage contrastive learning," *arXiv preprint arXiv:2308.03281*, 2023.