

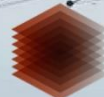
# LLMs4OL 2024: The 1st Large Language Models for Ontology Learning Challenge at the 23rd ISWC 11–15 November 2024, Baltimore, Maryland, USA

## Editors

Hamed Babaei Giglou

Jennifer D'Souza

Sören Auer



TIB LEIBNIZ-INFORMATIONSZENTRUM  
TECHNIK UND NATURWISSENSCHAFTEN  
UNIVERSITÄTSBIBLIOTHEK



## Open Conference Proceedings

Open Conference Proceedings (OCP) is an open series dedicated to publish proceedings from various conferences, workshops, symposia, and other academic events.

ISSN (online): 2749-5841



Open Conference Proceedings (OCP) is published by TIB Open Publishing (Technische Informationsbibliothek, Welfengarten 1 B, 30167 Hannover).



All contributions are distributed under the Creative Commons Attribution 4.0 International License.

## Volume 4

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

11 -15 November 2024, Baltimore, Maryland, USA

<b>Preface</b>		
Babaei Giglou et al.	Preface for LLMs4OL 2024: The 1st Large Language Models for Ontology Learning Challenge at the 23rd ISWC	1
<b>LLMs4OL 2024 Task Overview</b>		
Babaei Giglou et al.	LLMs4OL 2024 Overview: The 1st Large Language Models for Ontology Learning Challenge	3
Babaei Giglou et al.	LLMs4OL 2024 Datasets: Toward Ontology Learning with Large Language Models	17
<b>LLMs4OL 2024 Task Participant Papers</b>		
Kumar Goyal et al.	silp_nlp at LLMs4OL 2024 Tasks A, B, and C: Ontology Learning through Prompts with LLMs	31
Sanaei et al.	Phoenixes at LLMs4OL 2024 Tasks A, B, and C: Retrieval Augmented Generation for Ontology Learning	39
Peng et al.	RWTH-DBIS at LLMs4OL 2024 Tasks A and B: Knowledge-Enhanced Domain-Specific Continual Learning and Prompt-Tuning of Large Language Models for Ontology Learning	49
Atezong Ymele and Jiomekong	TSOTSALearning at LLMs4OL Tasks A and B : Combining Rules to Large Language Model for Ontology Learning	65
Barua et al.	DaSeLab at LLMs4OL 2024 Task A: Towards Term Typing in Ontology Learning	77
Phuttaamart et al.	The Ghost at LLMs4OL 2024 Task A: Prompt-Tuning-Based Large Language Models for Term Typing	85
Abi Akl	DSTI at LLMs4OL 2024 Task A: Intrinsic Versus Extrinsic Knowledge for Type Classification: Applications on WordNet and GeoNames Datasets	93
Hashemi et al.	SKH-NLP at LLMs4OL 2024 Task B: Taxonomy Discovery in Ontologies Using BERT and LLaMA 3	103

<https://doi.org/10.52825/ocp.v4i>

### **Editors**

Hamed Babaei Giglou, TIB Leibniz Information Centre for Science and Technology, Hannover, Germany

Dr. Jennifer D'Souza, TIB Leibniz Information Centre for Science and Technology, Hannover, Germany

Prof. Dr. Sören Auer, TIB Leibniz Information Centre for Science and Technology, Hannover, Germany

### **Review process**

The contributions undergo a review process via the EasyChair Platform. Each paper was reviewed at least by three reviewers to ensure quality.

### **Financing**

Funding for the publication of this volume is provided by the NFDI4DataScience initiative (DFG, German Research Foundation, Grant ID: 460234259) and the SCINEXT project (BMBF, German Federal Ministry of Education and Research, Grant ID: 01IS22070).

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

Hamed Babaei Giglou<sup>✉</sup>, Jennifer D'Souza<sup>✉</sup>, and Sören Auer<sup>✉</sup>

TIB Leibniz Information Centre for Science and Technology, Hannover, Germany

{hamed.babaei, jennifer.dsouza, auer}@tib.eu

We are pleased to present the proceedings of the “1st Large Language Models for Ontology Learning Challenge (LLMs4OL 2024)”, held at the 23rd International Semantic Web Conference (ISWC). This challenge marks a significant advancement in utilizing Large Language Models (LLMs) for Ontology Learning (OL)—a key Semantic Web component that facilitates the automatic extraction of structured knowledge from unstructured data. The challenge features three main tasks: *Term Typing* (identifying categories for terms), *Taxonomy Discovery* (uncovering hierarchical relationships), and *Non-Taxonomic Relation Extraction* (identifying other meaningful relationships between terms). Each task is designed to test different facets of ontology construction and to encourage the exploration of innovative techniques. This challenge seeks to foster collaboration, inspire innovation, and expand the capabilities of LLMs in OL. The proceedings include a collection of innovative solutions and insights from global participants, highlighting the crucial role of LLMs in enhancing the web with structured knowledge. We believe the outcomes of this challenge will propel further advancements in OL and its applications on the semantic web.

We would like to extend our gratitude to all the participants for their invaluable contributions, which their solutions and dedication have greatly enriched this challenge. Our sincere thanks also go to the conference organizers and committee for their efforts in hosting this event. We are deeply appreciative of the reviewers for their evaluations and feedback. Their reviews have been instrumental in enhancing the quality of the submissions. We would like to specifically acknowledge:

- *Dr. Amin Keshavarzi* (Postdoctoral Researcher, L3S & TIB, Germany)
- *Mostafa Rahgouy* (Lead Project Coordinator, Auburn University, USA)
- *Sahar Tahmasebi* (Doctoral Researcher, TIB, Germany)
- *Milad Molazadeh Oskuee* (Lead NLP Researcher, Iran)
- *Aida Usmanova* (Doctoral Researcher, Leuphana Universität Lüneburg, Germany)
- *Emetis Niazmand* (Doctoral Researcher, TIB, Germany)

Lastly, we would like to express our gratitude to "*Xenia Felice van Edig*" and "*Karolina Linerová*" from the *TIB Open Publishing*, for their support in bringing these proceedings online.

We also would like to acknowledge that the 1st LLMs4OL Challenge @ ISWC 2024 jointly supported by the [NFDI4DataScience initiative](#) (DFG, German Research Foundation, Grant ID: 460234259) and the [SCINEXT project](#) (BMBF, German Federal Ministry of Education and Research, Grant ID: 01IS22070).

# LLMs4OL 2024 Overview: The 1st Large Language Models for Ontology Learning Challenge

Hamed Babaei Giglou<sup>1</sup>, Jennifer D'Souza<sup>1</sup>, and Sören Auer<sup>1</sup>

TIB Leibniz Information Centre for Science and Technology, Hannover, Germany  
{hamed.babaei, jennifer.dsouza, auer}@tib.eu

\*Correspondence: Hamed Babaei Giglou, [hamed.babaei@tib.eu](mailto:hamed.babaei@tib.eu)

**Abstract:** This paper outlines the LLMs4OL 2024, the first edition of the Large Language Models for Ontology Learning Challenge. LLMs4OL is a community development initiative collocated with the 23rd International Semantic Web Conference (ISWC) to explore the potential of Large Language Models (LLMs) in Ontology Learning (OL), a vital process for enhancing the web with structured knowledge to improve interoperability. By leveraging LLMs, the challenge aims to advance understanding and innovation in OL, aligning with the goals of the Semantic Web to create a more intelligent and user-friendly web. In this paper, we give an overview of the 2024 edition of the LLMs4OL challenge<sup>1</sup> and summarize the contributions.

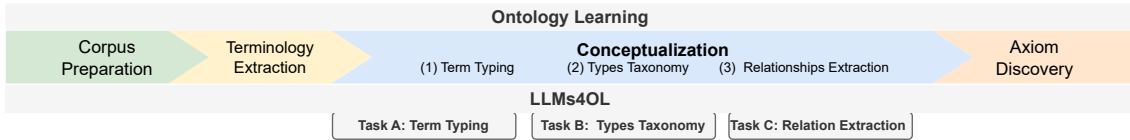
**Keywords:** LLMs4OL Challenge, Ontology Learning, Large Language Models

## 1 Introduction

The Semantic Web aims to enrich the current web with structured knowledge and metadata, enabling enhanced interoperability and understanding across diverse systems. At the core of this endeavor is Ontology Learning (OL), a process that automates the extraction of structured knowledge from unstructured data [1], essential for constructing dynamic ontologies that underpin the Semantic Web. The emergence of Large Language Models (LLMs) like GPT-3 [2] and GPT-4 [3] has revolutionized natural language processing (NLP), demonstrating remarkable performance across tasks such as language translation, question answering, and text generation. These models are particularly adept at crystallizing existing textual knowledge from a vast array of sources, making them potentially valuable for OL, where the goal is to extract a shared conceptualization of concepts and relationships from diverse inputs [4]. The introduction of LLMs has thus opened up new avenues of research, including the exploration of their potential in automating the OL process.

In our prior work published in the ISWC 2023 research track proceedings titled “LLMs4OL: Large Language Models for Ontology Learning” [5], marked a notable direction towards employing LLMs in OL, demonstrating their potential in automating knowledge acquisition and representation for the Semantic Web. Based on this research, the **The 1st Large Language Models for Ontology Learning Challenge at**

<sup>1</sup><https://sites.google.com/view/llms4ol>



**Figure 1.** The LLMs4OL task paradigm is an end-to-end framework for ontology learning. The three OL tasks that empirically validated in the LLMs4OL 2024 challenge, based on our prior research [5], are depicted within the blue arrow, aligned with the greater LLMs4OL paradigm.

**the 23rd ISWC 2024 (1st LLMs4OL Challenge @ ISWC 2024)** was introduced as a call for community development. With the LLMs4OL challenge, we aimed to catalyze community-wide engagement in validating and expanding the use of LLMs in OL. This initiative is poised to advance our comprehension of LLMs’ roles within the Semantic Web, encouraging innovation and collaboration in developing scalable and accurate ontology learning methods.

LLMs4OL consists of three OL tasks, *Task A – Term Typing*, *Task B – Taxonomy Discovery*, and *Task C – Non-Taxonomic Relation Extraction*. While participation in all three tasks in the LLMs4OL 2024 challenge is stipulated as desirable, but not mandatory. Thus participants choose to enroll only in Task A or B or C, or Task A and B, or Task A and C, or Task B and C. Furthermore, participants are required to implement LLM-based solutions, we did not impose any restrictions on the LLM prompting methods. For instance, they can choose to bring in additional context information from the World Wide Web to enrich the training and test instances. To thoroughly explore the potential of LLMs for OL, we structured the challenge around two distinct evaluation phases: (1) *Few-shot testing phase* and (2) *Zero-shot testing phase*. Through this work, we aim to contribute to the ongoing discourse on the capabilities of LLMs, particularly in the context of OL, and to provide insights into their potential for enhancing the Semantic Web. Thus, in the remainder of this paper, we detail the challenge tasks, what LLMs are being used, participant contributions, and findings.

## 2 LLMs4OL 2024 Tasks

In the LLMs4OL 2024 challenge, we have organized three main tasks which are centered around the ontology primitives [6] that comprise the following: **1.** a set of strings that describe terminological lexical entries  $L$  for conceptual types; **2.** a set of conceptual types  $T$ ; **3.** a taxonomy of types in a hierarchy  $H_T$ ; **4.** a set of non-taxonomic relations  $R$  described by their domain and range restrictions arranged in a heterarchy of relations  $H_R$ ; and **5.** a set of axioms  $A$  that describe additional constraints on the ontology and make implicit facts explicit.

To address these primitives, the tasks for OL [7] are: 1) Corpus preparation – collecting source texts for building ontology. 2) Terminology extraction – extracting relevant terms from the texts. 3) Term typing – grouping similar terms into conceptual types. 4) Taxonomy construction – establishing “is-a” hierarchies between types. 5) Relationship extraction – extracting semantic relationships beyond “is-a” between types. 6) Axiom discovery – finding constraints rules for the ontology. These tasks constitute the LLMs4OL task paradigm as depicted in Figure 1. Assuming the corpus preparation step is done by reusing ontologies publicly released in the community, we introduced the following three main tasks for the first edition of the LLMs4OL challenge.



**Table 1.** LLMs4OL 2024 challenge, subtasks, domains, number of participants per subtasks, and evaluation phases.

Task	SubTask	Domain	Participants	Phase
A	A.1 - WordNet	lexicosemantics	7	Few-shot
	A.2 - GeoNames	geographical locations	5	
	A.3 - UMLS - NCI	biomedical	5	
	A.3 - UMLS - MEDCIN		4	
	A.3 - UMLS - SNOMEDCT_US		4	
	A.4 - GO - Biological Process	biological	5	
	A.4 - GO - Cellular Component		5	
	A.4 - GO - Molecular Function		5	
	A.5 - DBO	general knowledge	2	Zero-shot
A.6 - FoodOn	food	2		
B	B.1 - GeoNames	geographical locations	5	Few-shot
	B.2 - Schema.org	web content types	3	
	B.3 - UMLS	biomedical	3	
	B.4 - GO	biological	1	Zero-shot
	B.5 - DBO	general knowledge	2	
	B.6 - FoodOn	food	1	
C	C.1 - UMLS	biomedical	2	Few-shot
	C.2 - GO	biological	0	
	C.3 - FoodOn	food	0	Zero-shot

## 2.1 Task A – Term Typing

The Table 1 shows 10 subtasks for *Task A* across 6 distinct domains such as lexicosemantics, geographical locations, biomedical, biological, general knowledge, and food domains. This task is defined as "*discover the generalized type for a given lexical term*". For this task, for each ontology, participants are given training instances defined as following formalism.

$$f_{prompt}^{TaskA}(L) := [S?]. ([L], [T])$$

Where  $S$  is an optional context sentence (if available in the source ontology),  $L$  is the lexical term prompted for, and  $T$  is the conceptual term type. In the test phase, types are hidden, and participants predict them for given terms using their trained models.

## 2.2 Task B – Taxonomy Discovery

After grouping terms under a conceptual type, in Task B, the goal is for given types "*discover the taxonomic hierarchy between types*", where the hierarchy between types is defined with an "is-a" relationship. Participants receive training instances for 6 distinct subtasks (described in Table 1) as :

$$f_{prompt}^{TaskB}(a, b) := (T_a, T_b)$$

Where  $T_a$  is the parent (superclass) of  $T_b$ , and  $T_b$  is the child (subclass) of  $T_a$ . The goal is to train a system to correctly identify the taxonomy between type. The training dataset will include term types and taxonomically related type pairs. In the test phase, participants work with just term types and must use their trained models to identify correct taxonomic relationships (type pairs). The types for the training and test phases are mutually exclusive. Furthermore, for the testing phase participants are required to post-process their outputs to return type pairs that follow the order of superclass-subclass related types.

### 2.3 Task C – Non-Taxonomic Relation Extraction

Nonetheless, the "is-a" relations are not the only relations in ontologies. So, Task C aims to "identify non-taxonomic, semantic relations between types". Training instances are given for three subtasks *C.1 - UMLS*, *C.2 - GO*, and *C.3 - FoodOn* as:

$$f_{prompt}^{TaskC}(h, r, t) := (T_h, r, T_t)$$

Where,  $T_h$  and  $T_t$  are head and tail taxonomic types, respectively, and  $r$  is the non-taxonomic semantic relation between them, chosen from a predefined set  $R$  of semantic relations. Participants aimed to train a system to identify pairs of types, and then classify pairs of types into semantic relations. The training phase involves types, relations, and triples of semantic relations; the test phase requires applying the trained system to predict semantically related triples from given types and the set of relations.

The caveat here is that we do not expect participant systems to infer a semantic relation but rather establish semantically related types and classify their relation from a known set of predetermined relations. This implies that any manual ontology specification task predetermines which semantic relations hold for the given ontology. In an alternative scenario, where participants might have had to infer the semantic relation, we realize that the possibilities of semantic relations might have been rather vast. Hence we posit a more realistic task design by predetermining the possible set of semantic relations.

## 3 Evaluation

There are two main evaluation phases for the challenge, which are the following:

- **Few-shot testing phase.** Each ontology selected for system training will be divided into two parts: one part will be released for the training of the systems and another part will be reserved for the testing of systems in this phase.
- **Zero-shot testing phase.** New ontologies that are unseen during training will be introduced. The objective is to evaluate the generalizability and transferability of the LLMs developed in this challenge.

For evaluation, we used the challenge datasets [8] – available at challenge GitHub<sup>2</sup> repository – with *standard evaluation metrics* used for all tasks. Given  $\mathcal{G}(i)$  as a set of ground truth labels for sample  $i$ , and  $\mathcal{P}(i)$  as a set of predicted labels for sample  $i$ , the precision  $P$ , recall  $R$ , and F1-score  $F1$  are being calculated as follows:

$$P = \frac{\sum_i |\mathcal{G}(i) \cap \mathcal{P}(i)|}{\sum_i |\mathcal{P}(i)|}, \quad R = \frac{\sum_i |\mathcal{G}(i) \cap \mathcal{P}(i)|}{\sum_i |\mathcal{G}(i)|}, \quad F1 = \frac{2 \times P \times R}{P + R}$$

With precision, we assessed the percentage of the returned related pairs, while recall was used to measure the proportion of correct pairs that were accurately retrieved. In the end, the F1-score was calculated as the harmonic mean of precision and recall, serving as a comparison metric for the participants' submissions. We used Codalab<sup>3</sup> [9] submission platform to organize participants submissions and scoring.

## 4 Participant Systems and Results

The LLMs4OL 2024 challenge has inspired diverse solutions, showcasing the growing potential of LLMs for OL tasks. Using the Codalab submissions platform, for this

<sup>2</sup><https://github.com/HamedBabaei/LLMs4OL-Challenge-ISWC2024>

<sup>3</sup><https://codalab.lisn.upsaclay.fr/competitions/19547>

**Table 2.** LLMs4OL 2024 challenge participants methods. \* refers to the subtask that did not make the submission to the leaderboard but was reported in the paper. MF refers to "Molecular Function", CC refers to "Cellular Component", and BF refers to "Biological Process". NCI, SNOMEDCT\_US, and MEDCIN are from "UMLS".

Team Name	LLM of Use	Approach	Code	A.1 - WordNet	A.2 - GeoNames	A.3 - NCI	A.3 - MEDCIN	A.3 - SNOMEDCT_US	A.4 - GO - BP	A.4 - GO - CC	A.4 - GO - MF	A.5 - DBO	A.6 - FoodOn	B.1 - GeoNames	B.2 - Schema.org	B.3 - UMLS	B.4 - GO	B.5 - DBO	B.6 - FoodOn	C.1 - UMLS	C.2 - GO	C.3 - FoodOn
DSTI [10]	Flan-T5 GTE-Large	Fine-tuning RAG	🔗	✓	*																	
DaSeLab [11]	GPT-3.5-Turbo	Fine-tuning	🔗	✓	✓	✓	✓	✓														
RWTH-DBIS [12]	GPT-3.5-Turbo LLaMA-3-8B	Prompting Fine-Tuning	🔗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓							
SKH-NLP [13]	LLaMA-3-70B Sentence-BERT	Prompting Fine-Tuning	🔗											✓								
TheGhost [14]	BLOOM-1B7 BLOOM-3B BLOOM- 7B1 LLaMA-7B LLaMA-2-7B LLaMA-3-8B BioMistral-7B OpenBioLLM-8B	Prompt-Tuning	🔗	✓	✓	✓	✓	✓	✓	✓	✓											
slip_nlp [15]	GPT-4o Mixtral-8x7B LLaMA-3-8B BERT Sentence-BERT	Prompting Fine-Tuning ML	🔗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					✓	
Phoenixes [16]	Mistral-7B Sentence-BERT	RAG	🔗	✓	✓				✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓		
TSOTSALearning [17]	GPT-4 BERT	RAG Rules	🔗	✓	✓				✓	✓	✓		✓									

challenge we set a limit of 10 submissions per day and a total of 30 submissions per subtask. We received 272 total submissions from 14 participants. In final, this challenge attracted the interest of the final eight research teams, as demonstrated by the various approaches they submitted for the subtasks. Each subtask of the competition depicted a rigorous field inherent to OL, which helped facilitate breakthroughs in finding generalized types (Task A), identifying taxonomic hierarchies (Task B), and extracting non-taxonomic relations (Task C), further scaffolding future AI advancements. Notably, teams employed varied strategies to tackle subtasks, such as fine-tuning, prompt-tuning, and retrieval-augmented generation (RAG). These approaches were used to analyze OL tasks across domains like lexicosemantics, geographical locations, biomedical concepts, and more (see Table 1 for subtasks and domains involved in this challenge). The summary of explored LLMs and subtasks are presented in Table 2 and in the following we will detail contributions and findings.

#### 4.1 Participants Contributions

The results for Task A are presented in Table 3, for Task B in Table 5, and for Task C in Table 4.

**DSTI [10].** DSTI fine-tuned Flan-T5-Small [18] model for *SubTasks A.1 - WordNet* and *A.2 - GeoNames*. Obtained F1-score of 0.9716 for *SubTask A.1* and ranked as a second team. But for GeoNames they did not submit the model to the leaderboard due to the larger nature of GeoNames dataset that required more computational resources. They introduced two approaches for OL. The first approach is fine-tuning LLMs using the zero-shot prompting method, the second approach is using a RAG pipeline using the General Text Embeddings (GTE)-Large [19] model as a retriever and fine-tuned LLM as a retriever. Due to the computational resources they preferred to use the Flan-T5-small model, and the results showed the effectiveness of their approach.

**Table 3.** Task A - Term Typing Results for SubTasks

SubTask	Team Name	F1-Score	Precision	Recall
A.1 (FS) - WordNet	TSOTSALearning	<b>0.9938</b>	<b>0.9938</b>	<b>0.9938</b>
	DSTI	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
	Phoenixes	0.8158	0.7689	0.8687
A.2 (FS) - GeoNames	DaseLab	<b>0.5906</b>	0.5906	<b>0.5906</b>
	Silp_nlp	0.4433	<b>0.7503</b>	0.3146
	RWTH-DBIS	0.4355	0.4355	0.4355
	TSOTSALearning	0.2937	0.2937	0.2937
	TheGhost	0.1489	0.1461	0.1519
A.3 (FS) - UMLS - NCI	DaseLab	<b>0.8249</b>	0.8161	<b>0.8340</b>
	Silp_nlp	0.6974	<b>0.8792</b>	0.5779
	TheGhost	0.5370	0.4450	0.6769
	RWTH-DBIS	0.1691	0.1821	0.1579
	Phoenixes	0.0737	0.0562	0.1070
A.3 (FS) - UMLS - MEDCIN	Silp_nlp	<b>0.9382</b>	<b>0.9591</b>	0.9181
	DaseLab	0.9373	0.9379	<b>0.9366</b>
	TheGhost	0.5328	0.4183	0.7336
	RWTH-DBIS	0.4566	0.4607	0.4526
A.3 (FS) - UMLS - SNOMEDCT_US	DaseLab	<b>0.8829</b>	<b>0.8810</b>	<b>0.8848</b>
	Silp_nlp	0.7552	0.8583	0.6742
	TheGhost	0.5275	0.4266	0.6910
	RWTH-DBIS	0.4747	0.4888	0.4613
A.4 (FS) - GO - Cellular Component	Silp_nlp	<b>0.2726</b>	<b>0.4279</b>	0.2000
	RWTH-DBIS	0.2178	0.1846	<b>0.2656</b>
	TheGhost	0.1877	0.1653	0.2171
	TSOTSALearning	0.0638	0.0767	0.0545
	Phoenixes	0.0158	0.0124	0.0217
A.4 (FS) - GO - Biological Process	Silp_nlp	<b>0.2691</b>	<b>0.4006</b>	<b>0.2026</b>
	TheGhost	0.1025	0.0964	0.1095
	RWTH-DBIS	0.0881	0.0693	0.1207
	TSOTSALearning	0.0648	0.0806	0.0542
	Phoenixes	0.0319	0.0214	0.0622
A.4 (FS) - GO - Molecular Function	Silp_nlp	<b>0.2970</b>	<b>0.4185</b>	<b>0.2302</b>
	RWTH-DBIS	0.1418	0.1670	0.1231
	TheGhost	0.1270	0.1278	0.1261
	TSOTSALearning	0.0910	0.1072	0.0790
	Phoenixes	0.0700	0.0485	0.1256
A.5 (ZS) - DBO	RWTH-DBIS	<b>0.4270</b>	<b>0.4270</b>	<b>0.4270</b>
	Silp_nlp	0.3009	0.3009	0.3009
A.6 (ZS) - FoodOn	RWTH-DBIS	<b>0.8068</b>	<b>0.8068</b>	<b>0.8068</b>
	Silp_nlp	0.7278	0.7278	0.7278

**RWTH-DBIS [12].** This team participated in tasks A and B (12 subtasks in total). For both tasks, they proposed a domain-specific continual training, fine-tuning, and knowledge-enhanced prompt-tuning approach. The models are firstly enriched with conceptual information related to terms and types. This is followed by CausalLM man-

ner and task-specific fine-tuning using LLaMA-3-8B [20]. The proposed approach performs well on several subtasks, showcasing that incorporating domain-specific information and providing a list of classification types enhances inference performance. They concluded that in Task A, GPT-3.5-Turbo [21] outperformed fine-tuned open-source LLM, and incorporating domain-specific information and providing a list of types at prompt significantly enhances the performance.

**DaSeLab [11].** The DaSeLab team participated in *UMLS*, *GeoNames*, and *WordNet* subtasks. This team approach is based on fine-tuning a GPT-3.5-Turbo model. The result of fine-tuning on *UMLS* and *GeoNames* domains showed that fine-tuning of such model can achieve superior performance. The DaSeLab ranked first place in *NCI* (0.8249), *GeoNames* (0.5906), and *SNOMEDCT\_US* (0.8829) subtasks (scores inside practices are F1-scores).

**TheGhost [14].** The TheGhost team investigated a variety of LLMs with a prompt-tuning approach. They are the first team in the challenge that explored 11 LLMs (the LLM list depicted in Table 2) for 8 subtasks of term typing tasks within a few-shot testing evaluation scenario. They showed the viability of soft prompt tuning for OL and the challenge of imbalanced class prompt tuning. Their finding supports the complexity of geographical and biological domains at the term typing task of OL.

**silp\_nlp [15].** The silp\_nlp team participated in all three tasks with a total of 15 subtasks. They ranked in first place in several subtasks including *A.3 (FS) - UMLS - MEDCIN* (0.9382), *A.4 (FS) - GO - Cellular Component* (0.2726), *A.4 (FS) - GO - Biological Process* (0.2691), *A.4 (FS) - GO - Molecular Function* (0.2970), *B.2 (FS) - Schema.org* (0.6157), *B.3 (FS) - UMLS*, *B.5 (FS) - DBO* (0.2109), and *C.1 (FS) - UMLS* (0.0783). They employed several machine learning techniques, such as Random Forest, Logistic Regression, and XGBoost, alongside advanced generative models like LLaMA-3-8B, Mixtral [22], and GPT-4o [3]. The results revealed that prompt-based methods were effective in some domains but not universally applicable. Notably, Random Forest models excelled in subtasks A.1 through A.4, while GPT-4o dominated the zero-shot tasks A.5 and A.6, as well as relation extraction tasks B and C. This team obtained in first-place in six subtasks and second place in five subtasks.

**TSOTSALearning [17].** The TSOTSALearning team focused on LLMs such as BERT [23] and GPT-4. Through experimentation on *SubTask A.1 - WordNet* dataset, they achieved an F1-score of 0.9264 with GPT-4, but significantly improved results when they combined BERT with rule-based strategies, leading to an F1-score of 0.9938 and ranked first place in *WordNet* dataset. Their findings showed the importance of incorporating rules into LLMs for enhanced accuracy in OL. However, they highlight the challenge of identifying appropriate rules, suggesting that future work should focus on automating rule detection and integrating them seamlessly into LLMs. The *WordNet* dataset is being considered as a low number of types and having a higher number of types makes it challenging to obtain highly accurate rules.

**SKH-NLP [13].** Team SKH-NLP participated in *SubTask B.1 - GeoNames*, where they developed a fine-tuning approach using the LLaMA-3-70B and BERT-Large [24]. This team obtained the first place in *SubTask B.1 - GeoNames* with an F1-score of 0.6557. Their comprehensive analysis demonstrates that BERT-Large, when fine-tuned, achieves performance close to the larger LLaMA-3-70B model.

**Phoenixes [16].** The Phoenixes team explored the application of a Retrieval Augmented Generation (RAG) approach within the 12 subtasks of the challenge. They introduced a promising RAG-specific formulation over all three tasks of OL, where a

**Table 4.** Task B - Taxonomy Discovery Results for SubTasks

SubTask	Team Name	F1-Score	Precision	Recall
B.1 (FS) - GeoNames	SKH-NLP	<b>0.6557</b>	<b>0.6318</b>	<b>0.6814</b>
	RWTH-DBIS	0.3409	0.2400	0.5882
	Silp_nlp	0.0830	0.0446	0.5931
	TSOTSA Learning	0.0104	0.0052	0.5294
	Phoenixes	0.0036	0.0019	0.0294
B.2 (FS) - Schema.org	Silp_nlp	<b>0.6157</b>	0.4578	<b>0.9396</b>
	RWTH-DBIS	0.5733	<b>0.5475</b>	0.6016
	Phoenixes	0.0155	0.0079	0.3901
B.3 (FS) - UMLS	Silp_nlp	<b>0.3544</b>	<b>0.4118</b>	0.3111
	Phoenixes	0.0960	0.0550	0.3778
	RWTH-DBIS	0.0491	0.0257	<b>0.5556</b>
B.4 (FS) - Gene Ontology (GO)	Phoenixes	0.0164	0.0180	0.0149
B.5 (FS) - DBpedia Ontology (DPO)	Silp_nlp	<b>0.2109</b>	0.1412	0.4164
	Phoenixes	0.0164	0.0180	0.0149
B.6 (ZS) - Food Ontology (FoodOn)	Phoenixes	0.0308	0.0243	0.0420

**Table 5.** Task C - Non-Taxonomic Relation Extraction Results for SubTasks

SubTask	Team Name	F1-Score	Precision	Recall
C.1 (FS) - UMLS	Silp_nlp	0.0783	<b>0.0494</b>	<b>0.1888</b>
	Phoenixes	0.0273	0.0433	0.0199

RAG system with minor changes was developed for both tasks A and B, later can be used as a two-step approach for task C. Task C consists of the following steps: Step 1 – runs the Task B approach for finding child-parent pairs and step 2 – applying the Task A approach for assigning the relations to the pairs. They incorporated Mistral-7B [25] as LLM and Dense Passage Retrieval (DPR) [26] model as the retriever model in the RAG framework. However, their results in both zero-shot and few-shot fall shorter than the fine-tuned models and this suggests that still fine-tuning is the key to obtain a high performance within OL.

## 4.2 Large Language Models

The participants in the challenge utilized a diverse array of LLMs, each bringing distinct strengths to the tasks. We detailed a breakdown of the key strengths of the prominent LLMs used.

**GPT FAMILY** – GPT-3.5-Turbo, GPT-4, and GPT-4o: GPT based LLMs, developed by OpenAI, are renowned for their advanced natural language understanding and generation capabilities. These models excel in context comprehension and can handle a variety of queries effectively, making them particularly suitable for tasks that require deep semantic understanding and detailed generation. Their ability to generalize from a wide range of training data allows them to perform well across various knowledge domains relevant ontologies [5], [27]. GPT-3.5-Turbo was a popular choice among participants, with teams such as DaSeLab, RWTH-DBIS, and silp\_nlp using the model and demonstrating its high adaptability and effectiveness across the various challenge sub-tasks. Furthermore, GPT-4 and GPT-4o as more advanced models over GPT-3, were explored by the teams: TSOTSA Learning and *silp\_nlp*.

**LLAMA FAMILY** – LLaMA-7B, LLaMA-2-7B, LLaMA-3-8B, and LLaMA-3-70B: The LLaMA models were another prominent choice among participants. With models like LLaMA-2 and LLaMA-3 featured by TheGhost, RWTH-DBIS, SKH-NLP, and silp\_nlp, their popularity stems from their open-source, efficiency, and scalability. These models' strengths in handling large-scale data and intricate details made them well-suited for comprehensive multi-dimensional data interpretation.

**BLOOM FAMILY** – BLOOM-1B7, BLOOM-3B, and BLOOM-7B1: BLOOM [28] models, featured in our original research work [5], gained traction due to their open-access nature and collaborative development. TheGhost, in particular, utilized a range of BLOOM models for their flexibility and multilingual capabilities.

**BIOMEDICAL FAMILY** – BioMistral-7B and OpenBioLLM-8B: BioMistral-7B [29], as a domain-specific fine-tuned variant of Mistral-7B, and OpenBioLLM-8B [30], as a domain-specific fine-tuned variant of LLaMA-3-8B, were utilized for their domain-specific strengths in biomedical contexts. TheGhost's use of these models highlights their importance in tasks requiring detailed biomedical terminology and concepts, emphasizing their significance in the specialized subfields of the challenge.

**MISTRAL FAMILY** – Mistral-7B and Mixtral-8x7B: Mistral-7B, part of the Mistral family of models, was noted for its performance in the challenge by teams like Phoenixes and TheGhost. Moreover, Mixtral-8x7B was utilized by the team silp\_nlp.

**OTHERS** – Flan-T5, GTE-Large, Sentence-BERT, and DPR: Flan-T5 and GTE-Large were chosen for their adaptability and fine-tuning capabilities. DSTI recognized their potential in fine-tuning and handling diverse NLP tasks efficiently when there are limited computational resources. Sentence-BERT was prominently used for tasks involving semantic similarity and sentence-level embeddings. Its popularity among participants like SKH-NLP and Phoenixes. Phoenixes used DPR for the retrieval model of the RAG approach.

### 4.3 Trade-offs Between Precision and Recall

Across the tasks, a clear trend emerges among the participating teams. Teams like silp\_nlp often exhibit high precision but lower recall, particularly in subtasks related to GO and UMLS ontologies. This suggests that while silp\_nlp is adept at avoiding false positives and making accurate predictions, it frequently misses relevant instances, indicating a more conservative approach. However, teams such as RWTH-DBIS and Phoenixes display a different trend, where recall is relatively higher than precision. These teams retrieve a larger number of relevant results but at the cost of precision, indicating that they tend to capture a broad set of possible answers, including many false positives. Their approach may be useful in tasks where coverage is prioritized over accuracy, but it also introduces challenges in filtering out noise.

Teams that manage to balance both precision and recall, such as DaSeLab and SKH-NLP, stand out for their well-rounded performance. These teams perform consistently across different subtasks by finding a middle ground between retrieving enough relevant results and minimizing false positives. DaSeLab, for example, shows balanced performance across multiple subtasks, especially in UMLS-related tasks, suggesting a more effective strategy. Meanwhile, SKH-NLP stands out in the GeoNames taxonomy discovery task, where it achieves high precision and recall, demonstrating its capability to capture relevant information without sacrificing accuracy.

In more challenging tasks, such as non-taxonomic relation extraction, the disparity between precision and recall becomes particularly pronounced. For example, both `silp_nlp` and `Phoenixes` struggle, with `silp_nlp` showing low precision but managing to retrieve more relevant results than `Phoenixes`, which has very low recall. This suggests that these tasks may require more sophisticated models or techniques to achieve higher performance. Overall, the results reflect that teams vary significantly in how they prioritize precision and recall, depending on the specific subtask, with some teams excelling in precision-oriented tasks while others show better results in recall-sensitive subtasks.

## 5 Discussion

**Performance Analysis.** As the participating teams navigated through the zero-shot and few-shot testing phases of the LLMs4OL 2024 challenge, notable variations in performance underscored the importance of model adaptability and data-specific adjustments. Few-shot tasks, particularly those involving geographical, biological, and biomedical domains, highlighted the critical need for specialized model tuning and the strategic use of training data to achieve high precision and recall rates. This indicates that achieving optimal performance in real-world ontology challenges requires not only selecting the right LLMs but also fine-tuning them to align with the specific characteristics of the domains and tasks at hand. Additionally, studies show that for Task A, even smaller models like `Flan-T5-Small` with 80M parameters can perform well when there are fewer types. However, as the number of types increases, larger models, such as those with 7B parameters, tend to perform better. One reason for the popularity of 7B models is that Parameter-Efficient Fine-Tuning (PEFT) [31] fine-tuning requires less memory compared to traditional fine-tuning methods. Many participants also incorporated external knowledge, such as type definitions, synthesis data using LLMs, or general knowledge graphs (KGs) to build answer sets. These strategies have demonstrated a positive impact on fine-tuning performance.

**Complexity Across Domains and Tasks.** The results indicated that certain domains and tasks, such as biomedical term typing and non-taxonomic relation extraction, were more challenging than others. The variation in performance across tasks, particularly in relation to term complexity (e.g., Gene Ontology), highlights the complexity of certain knowledge domains. This still requires specialized approaches. The `Phoenixes` (on all three tasks) and `DSTI` (on task A only) teams introduced a formulation based on Retrieval-Augmented Generation (RAG) approaches with success, indicating that combining LLM generation capabilities with retrieval mechanisms can enhance accuracy in OL tasks. This approach is particularly suitable due to the hybrid framework with high adaptability to be extended with different components.

**Few-Shot and Zero-Shot Testing Phases.** While many models performed well in the few-shot phase, the zero-shot testing phase exposed limitations in the generalization capabilities of LLMs. Models like `GPT-3.5` and `GPT-4` demonstrated strong performance, but there were notable drops when transitioning from few-shot to zero-shot testing phases. More research is needed to improve the transferability and robustness of LLMs across unseen domains and ontologies.

**Task A vs Task C.** From a task perspective, Task C attracted only two teams, indicating it was perceived as highly challenging. Non-taxonomic relation extraction requires identifying complex relationships between terms that go beyond hierarchical (taxonomy-based) relations, which is a significantly more intricate task. Unlike sim-



ple is-a relationships, non-taxonomic relations are more diverse, context-dependent, and require a deeper understanding of the subject matter. Extracting these relations often involves dealing with ambiguous or implicit connections, requiring models to infer meanings that might not be explicit. This complexity might have discouraged more teams from participating, as success in this task requires advanced techniques, often combining deep semantic understanding with domain-specific knowledge. On the other hand, Task A, term typing, had much higher participation compared to Task C. This task involves classifying terms into predefined categories, a more familiar task for many researchers. Term typing is conceptually simpler because it involves assigning a label to a term, which is something that even general-purpose LLMs can do relatively well. There is a clear, finite set of categories or types, and many participants experimented with text classification approaches.

## 6 Conclusion

The 1st Large Language Models for Ontology Learning Challenge at ISWC 2024 has revealed the emerging potential of LLMs beyond previous studies of OL tasks. The diverse range of participant systems, including fine-tuning, prompt-tuning, and retrieval-augmented generation approaches, demonstrated how adaptable LLMs can be when handling complex ontological data across various domains. The integration of diverse LLMs like GPT-4o, GPT-3.5, LLaMA-3, and Mistral underscored the versatility of LLMs.

Through this challenge, key insights were garnered regarding the strengths and limitations of current LLMs for OL. Notably, while LLMs have shown a remarkable capacity to generalize across unseen tasks (as evidenced by their performance in few-shot and zero-shot scenarios), certain domains such as biomedical and geographical ontologies posed unique challenges, particularly in terms of class imbalance and complex taxonomies. These challenges opened pathways for future research, emphasizing the need for scalable LLM training and the refinement of prompt-based methods to handle highly specialized ontologies.

Moreover, the variety of approaches suggests that hybrid methods combining LLMs with domain-specific knowledge are particularly effective. Moving forward, research should focus on improving the interpretability and scalability of LLM-based OL systems to enable even more accurate and dynamic knowledge extraction. This challenge has laid the groundwork for expanding LLM capabilities in the context of the Semantic Web, fostering innovation and collaboration in building the next generation of intelligent web technologies.

## Data Availability Statement

The datasets supporting the findings of this article are publicly available and can be accessed via Zenodo at <https://doi.org/10.5281/zenodo.13851373>, or through the GitHub repository: <https://github.com/HamedBabaei/LLMs4OL-Challenge-ISWC2024>.

## Authors Contributions

**Hamed Babaei Giglou:** Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing - Original Draft, Writing – Review & Editing, Visualization.

**Jennifer D'Souza:** Conceptualization, Methodology, Investigation, Resources, Super-

vision, Project administration, Funding acquisition, Writing – Review & Editing, Visualization.

**Sören Auer:** Conceptualization, Methodology, Review & Editing, Supervision, Project administration, Funding acquisition.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

The 1st LLMs4OL Challenge @ ISWC 2024 jointly supported by the [NFDI4DataScience initiative](#) (DFG, German Research Foundation, Grant ID: 460234259) and the [SCINEXT project](#) (BMBF, German Federal Ministry of Education and Research, Grant ID: 01IS22070).

## References

- [1] A. Konys, "Knowledge repository of ontology learning tools from text," *Procedia Computer Science*, vol. 159, pp. 1614–1628, 2019.
- [2] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: [2005.14165 \[cs.CL\]](#).
- [3] OpenAI, J. Achiam, S. Adler, S. Agarwal, and *et al.*, *Gpt-4 technical report*, 2024. arXiv: [2303.08774 \[cs.CL\]](#). [Online]. Available: <https://arxiv.org/abs/2303.08774>.
- [4] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International journal of human-computer studies*, vol. 43, no. 5-6, pp. 907–928, 1995.
- [5] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," in *The Semantic Web – ISWC 2023*, T. R. Payne, V. Presutti, G. Qi, *et al.*, Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.
- [6] A. Maedche and S. Staab, "Ontology learning for the semantic web," *IEEE Intelligent systems*, vol. 16, no. 2, pp. 72–79, 2001.
- [7] N. F. Noy, D. L. McGuinness, *et al.*, *Ontology development 101: A guide to creating your first ontology*, 2001.
- [8] 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.
- [9] A. Pavao, I. Guyon, A.-C. Letournel, *et al.*, "Codalab competitions: An open source platform to organize scientific challenges," *Journal of Machine Learning Research*, vol. 24, no. 198, pp. 1–6, 2023. [Online]. Available: <http://jmlr.org/papers/v24/21-1436.html>.
- [10] H. Abi Akl, "Dsti at llms4ol 2024 task a: Intrinsic versus extrinsic knowledge for type classification, Applications on wordnet and geonames datasets," *Open Conference Proceedings*, vol. 4, Oct. 2024.
- [11] A. Barua, S. Saki Norouzi, and P. Hitzler, "Daselab at llms4ol 2024 task a: Towards term typing in ontology learning," *Open Conference Proceedings*, vol. 4, Oct. 2024.
- [12] Y. Peng, Y. Mou, B. Zhu, S. Sowe, and S. Decker, "Rwth-dbis at llms4ol 2024 tasks a and b, Knowledge-enhanced domain-specific continual learning and prompt-tuning of large language models for ontology learning," *Open Conference Proceedings*, vol. 4, Oct. 2024.

- [13] S. Hashemi, M. Karimi Manesh, and M. Shamsfard, "Skh-nlp at llms4ol 2024 task b: Taxonomy discovery in ontologies using bert and llama 3," *Open Conference Proceedings*, vol. 4, Oct. 2024.
- [14] T. Phuttaamart, N. Kertkeidkachorn, and A. Trongratsameethong, "The ghost at llms4ol 2024 task a: Prompt-tuning-based large language models for term typing," *Open Conference Proceedings*, vol. 4, Oct. 2024.
- [15] P. Kumar Goyal, S. Singh, and U. Shanker Tiwari, "Silp\_nlp at llms4ol 2024 tasks a, b, and c: Ontology learning through prompts with llms," *Open Conference Proceedings*, vol. 4, Oct. 2024.
- [16] M. Sanaei, F. Azizi, and H. Babaei Giglou, "Phoenixes at llms4ol 2024 tasks a, b, and c: Retrieval augmented generation for ontology learning," *Open Conference Proceedings*, vol. 4, Oct. 2024.
- [17] C. Ymele and A. Jiomekong, "Combining rules to large language model for ontology learning," *Open Conference Proceedings*, vol. 4, Oct. 2024.
- [18] H. W. Chung, L. Hou, S. Longpre, et al., *Scaling instruction-finetuned language models*, 2022. arXiv: [2210.11416](https://arxiv.org/abs/2210.11416) [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2210.11416>.
- [19] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, *Towards general text embeddings with multi-stage contrastive learning*, 2023. arXiv: [2308.03281](https://arxiv.org/abs/2308.03281) [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2308.03281>.
- [20] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, and et al., *The llama 3 herd of models*, 2024. arXiv: [2407.21783](https://arxiv.org/abs/2407.21783) [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2407.21783>.
- [21] OpenAI, *Openai gpt-3.5 api [gpt-3.5-turbo]*, <https://platform.openai.com/docs/models/gpt-3-5>, Available at: <https://platform.openai.com/docs/models/gpt-3-5>, 2024.
- [22] A. Q. Jiang, A. Sablayrolles, A. Roux, et al., *Mixtral of experts*, 2024. arXiv: [2401.04088](https://arxiv.org/abs/2401.04088) [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2401.04088>.
- [23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Dorr, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423). [Online]. Available: <https://aclanthology.org/N19-1423>.
- [24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: [1810.04805](https://arxiv.org/abs/1810.04805) [cs.CL]. [Online]. Available: <https://arxiv.org/abs/1810.04805>.
- [25] A. Q. Jiang, A. Sablayrolles, A. Mensch, et al., *Mistral 7b*, 2023. arXiv: [2310.06825](https://arxiv.org/abs/2310.06825) [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2310.06825>.
- [26] V. Karpukhin, B. Oğuz, S. Min, et al., *Dense passage retrieval for open-domain question answering*, 2020. arXiv: [2004.04906](https://arxiv.org/abs/2004.04906) [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2004.04906>.
- [27] H. B. Giglou, J. D'Souza, F. Engel, and S. Auer, *Llms4om: Matching ontologies with large language models*, 2024. arXiv: [2404.10317](https://arxiv.org/abs/2404.10317) [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2404.10317>.
- [28] B. Workshop, : T. L. Scao, A. Fan, and et al., *Bloom: A 176b-parameter open-access multilingual language model*, 2023. arXiv: [2211.05100](https://arxiv.org/abs/2211.05100) [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2211.05100>.

- [29] Y. Labrak, A. Bazoge, E. Morin, P.-A. Gourraud, M. Rouvier, and R. Dufour, *Biomistral: A collection of open-source pretrained large language models for medical domains*, 2024. arXiv: [2402.10373](https://arxiv.org/abs/2402.10373) [cs.CL].
- [30] M. S. Ankit Pal, *Openbiollms: Advancing open-source large language models for health-care and life sciences*, <https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B>, 2024.
- [31] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, *Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment*, 2023. arXiv: [2312.12148](https://arxiv.org/abs/2312.12148) [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2312.12148>.

# LLMs4OL 2024 Datasets: Toward Ontology Learning with Large Language Models

Hamed Babaei Giglou<sup>✉</sup>, Jennifer D'Souza<sup>✉</sup>, Sameer Sadruddin<sup>✉</sup>, and Sören Auer<sup>✉</sup>

TIB Leibniz Information Centre for Science and Technology, Hannover, Germany  
{hamed.babaei, jennifer.dsouza, sameer.sadruddin, auer}@tib.eu

\*Correspondence: Hamed Babaei Giglou, hamed.babaei@tib.eu

**Abstract:** Ontology learning (OL) from unstructured data has evolved significantly, with recent advancements integrating large language models (LLMs) to enhance various aspects of the process. The paper introduces the LLMs4OL 2024 datasets, developed to benchmark and advance research in OL using LLMs. The LLMs4OL 2024 dataset as a key component of the LLMs4OL Challenge, targets three primary OL tasks: Term Typing, Taxonomy Discovery, and Non-Taxonomic Relation Extraction. It encompasses seven domains, i.e. lexosemantics and biological functions, offering a comprehensive resource for evaluating LLM-based OL approaches. Each task within the dataset is carefully crafted to facilitate both Few-Shot (FS) and Zero-Shot (ZS) evaluation scenarios, allowing for robust assessment of model performance across different knowledge domains to address a critical gap in the field by offering standardized benchmarks for fair comparison for evaluating LLM applications in OL.

**Keywords:** Ontology Learning, Large Language Models, Dataset, LLMs4OL Challenge

## 1 Introduction

Ontologies have gained a lot of popularity and recognition in the semantic web because of their fine source of semantics and interoperability. The increase in unstructured data on the web has made the automated acquisition of ontology from unstructured text a most prominent research area. Recently, instead of handcrafting ontologies, the research trend is now shifting toward automatic ontology learning (OL) [1]. OL involves automatically identifying terms, types, relations, and potential axioms from textual information to construct an ontology [2].

Looking back to the history of OL research, until early 2002 [3], most OL approaches relied on seed words or existing base ontologies rather than building new ones from scratch. Later in 2003 [4] the natural language processing (NLP) technique showed promise for the extraction of new concepts. However, relation extraction for OL remained still challenging. Also, the prior domain knowledge of the base ontologies still was in the middle of the focus for OL. With progress in the field, in 2006 the concept of "ontology learning layer cake" [5] was introduced to organize and describe the different steps involved in the process of ontology learning from the text for real-life application

scenarios. The OL layer cake includes (from the bottom of the cake to the top), Terms, Synonyms, Concepts, Taxonomies, Relations, Rules, and Axioms. This reflects a progression from simpler to more complex and abstract forms, each step building on the results of the previous one. It provides a structured approach to understanding and automating the OL process. Later in 2011, Hazman et al.[6] studied various OL systems and categorized them into two categories, (1) *learning from unstructured data* and (2) *learning from semi-structured data*. They also pointed out that when human-based evaluation is not possible, carrying out five-level evaluations for OL is important, levels such as lexical, hierarchical, contextual, syntactic, and structural levels. Since 2011 and in 2018 survey of [7] showed that a hybrid approach comprising both linguistic and statistical techniques produces better ontologies. However, it is difficult to find the best technique amount approaches due to the domain of the studies. The trend was shifted toward statistical techniques for term extractions, however for relation extraction clustering methods were the most used ones. Moreover, the various evaluations of OL showed that human-based evaluation is the most reliable approach for evaluation.

Considering that most of the approaches in the field were based on statistical approaches or clustering models, the emergence of large language models (LLMs), offered a paradigm shift in OL since their characteristics justify OL as a studied for the first time within LLMs4OL paradigm [8]. One reason for this shift is the LLM's generation capabilities because they are being trained on extensive and diverse text, similar to domain-specific knowledge bases [9]. For the first time, in 2023 the LLMs4OL [8] paradigm was introduced that incorporates LLMs for three important tasks of OL as Term Typing, Taxonomy Discovery, and Non-Taxonomic Relation Extraction. Later, more researchers were involved in the OL tasks from different perspectives [10]–[13].

The current trend in the semantic web reveals a growing interest among researchers in utilizing LLMs [14]. A benchmark dataset is essential to assess the performance of OL approaches, particularly those involving LLMs, in a consistent and comparable manner. Without such benchmarks, it becomes difficult to evaluate progress and compare various methodologies effectively [13]. To address this gap, in this work, we introduce an LLMs4OL paradigm tasks dataset to bridge the gap in benchmark evaluation datasets specifically within the context of OL using LLMs. Our key contribution is the creation of the LLMs4OL dataset, aimed at facilitating consistent evaluation in this emerging field. For the first time, this dataset is introduced in the "*1st LLMs4OL Challenge @ ISWC 2024*" [15], a challenge organized at the prestigious International Semantic Web Conference (ISWC). The primary goal of the challenge is to provide a shared platform for researchers to benchmark their LLM-based OL approaches. By establishing this dataset and launching the LLMs4OL Challenge, we hope to encourage further research and innovation in OL with LLMs, ultimately enabling a more structured and fair comparison of different methods in this rapidly evolving area.

The LLMs4OL 2024 dataset addresses three OL tasks, which are known as primitive ontology construction tasks [16]. Considering,  $L$  as a lexical entries for conceptual type  $T$ , and  $H_T$  as a representation of taxonomy of types, and  $R$  as a non-taxonomic relations, the LLMs4OL tasks are defined as follows:

- **Task A – Term Typing:** For a given lexical term  $L$ , discover the generalized type  $T$ .
- **Task B – Taxonomy Discovery:** For a given set of generalized types  $T$ , discover the taxonomic hierarchical pairs  $(T_a, T_b)$  pairs, representing "is-a" relations.
- **Task C – Non-Taxonomic Relation Extraction:** For a given set of generalized types  $T$  and relations  $R$ , identify non-taxonomic, semantic relations between

types to form a  $(T_h, r, T_t)$  triplet, where  $T_h$  and  $T_t$  are head and tail taxonomic types with  $r \in R$ .

The LLMs4OL dataset is publicly available on GitHub<sup>1</sup>, providing easy access for researchers and practitioners in the field. The paper is organized as follows: Section 2 describes the domains that are being considered for benchmarking LLMs4OL and Section 3 investigates how ontologies are curated for OL. In section 4, we discuss the curated dataset. Finally, we conclude in Section 5

## 2 Ontological Resources and Domains of the Study

The *LLMs4OL 2024 datasets* support a variety of domains from lexosemantics to biomedical. Such variety supports the comprehensiveness of the studies within the *LLMs4OL 2024 Challenge*. In the following, we detail each ontology within the domains that we used for the construction of the LLMs4OL paradigm tasks dataset.

**Lexosemantics.** WordNet [17] is a large lexical database of English that serves as a rich ontology for NLP and other applications. It was developed at Princeton University and has become a widely used tool for understanding and representing the relationships between words. WordNet is divided into four main parts of speech, 1) Nouns: Concepts, entities, and objects. 2) Verbs: Actions, processes, or states of being. 3) Adjectives: Descriptive qualities or attributes. 4) Adverbs: Modifiers of verbs, adjectives, or other adverbs. Each part of speech has its own set of synsets and relationships, which helps in distinguishing the different meanings words can have when used in different grammatical contexts

**Geographical Locations.** The GeoNames [18] Ontology is a formal representation of geographical data that models geographic features, locations, and associated information. It is a crucial part of the Linked Open Data (LOD) cloud, providing a machine-readable format for geographic data to facilitate integration, querying, and sharing of geographic knowledge across different domains. GeoNames contains over 12 million geographical names and 9 million unique features such as cities, countries, rivers, mountains, lakes, etc. This makes GeoNames a rich ontology for further studies of LLMs4OL tasks.

**Biomedical.** The Unified Medical Language System (UMLS) [19] is a comprehensive biomedical ontology developed and maintained by the U.S. National Library of Medicine (NLM). It integrates various healthcare terminologies, coding systems, and ontologies to create a unified resource that supports NLP, biomedical data integration, and interoperability between different healthcare systems. UMLS Metathesaurus is a large database of biomedical concepts and terms that integrates many existing terminologies and coding systems. It consists of source vocabularies and includes well-known ontologies like SNOMEDCT\_US [20], NCI [21], and MEDCIN [22]. The SNOMEDCT\_US provides the core general terminology for the electronic health record. However, NCI covers vocabulary for cancer-related clinical care, translational and basic research, and public information and administrative activities. Moreover, the MEDCIN medical terminology encompasses symptoms, history, physical examination, tests, diagnoses, and therapies.

**Biological.** Gene Ontology (GO) [23] consortium is a major bioinformatics initiative that provides a standardized vocabulary to describe the functions, locations, and processes involving genes and gene products across different species. GO aims to unify

---

<sup>1</sup><https://github.com/HamedBabaei/LLMs4OL-Challenge-ISWC2024>

the representation of gene and gene product attributes, allowing researchers to consistently annotate biological data and make it easier to compare gene functions across organisms. GO provides a hierarchical structure to describe gene products in three key areas such as *Biological Process (BP)*, *Molecular Function (MF)*, and *Cellular Component (CC)*. The BP describes our knowledge of the biological domain in the larger processes accomplished by multiple molecular activities. The CC goes beyond molecular activities and considers only location, relative to cellular compartments and structures. MF describes activities that occur at the molecular level, such as "catalysis" or "transport".

**General Knowledge.** DBpedia [24] is a crowd-sourced initiative aimed at extracting structured data from content generated across various Wikimedia projects. This data forms an open knowledge graph (OKG) that is accessible to everyone on the Web. The DBpedia Ontology (DBO), as a cross-domain ontology, emerged from a community effort to use Wikipedia's most commonly used infoboxes to create a formal vocabulary for categorizing knowledge for more precise querying and data linking. Wikipedia articles, typically representing specific entities (e.g., people, places, or events), can be classified under one or more of these classes. As a result of this, the ontology is structured as a hierarchy of classes and properties that describe concepts and their relationships, resulting in 768 classes, which form a subsumption hierarchy with around 3,000 properties and contain approximately 4 million instances.

**Food.** Food Ontology (FoodOn) [25] is a consortium-driven project to build a comprehensive and easily accessible global farm-to-fork ontology about food, that accurately and consistently describes foods commonly known in cultures from around the world. The FoodOn as a food product terminology supports food security, safety, quality, production, distribution, and consumer health and convenience.

**Web Content Types.** Schema.org [26] vocabulary covers entities, relationships between entities, and actions, and can easily be extended through a well-documented extension model. The schemas are a set of 'types', each associated with a set of properties and the types are arranged in a hierarchy. Overall, schema.org consists of 806 Types, 1476 properties 14 datatypes, 90 enumerations, and 480 enumeration members.

### 3 Ontology Curation for LLMs4OL Tasks

We curated 6 ontologies comprising a total of 10 datasets for Task A, 6 ontologies for Task B, and 3 ontologies for Task C. The curated ontologies and processes are represented in Figure 1, which involves three steps, each corresponding to the specified tasks. In this section, we provide a brief overview of the curation process.

#### 3.1 WordNet Ontology – Task A

We utilized the WN18RR dataset, as introduced in [27]. For evaluation, we merged the test and validation sets, while the original training set was retained for model training. Additionally, we focused on four specific lexical term types  $T$ : nouns, verbs, adverbs, and adjectives. We also incorporated the sentences available in the WordNet dataset as additional context for the terms.



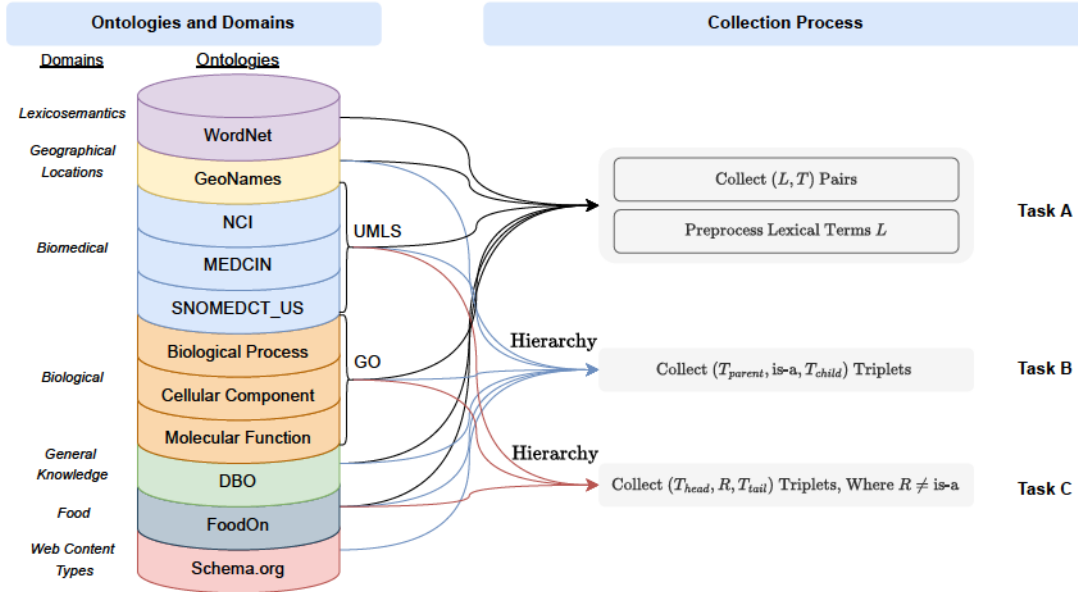


Figure 1. LLMs4OL 2024 Datasets Curation.

### 3.2 GeoNames Ontology – Tasks A and B

The GeoNames ontology encompasses all geographical locations worldwide. To narrow our focus, we first restricted the dataset to locations represented in English letters, resulting in a set of lexical terms  $L$ . GeoNames uses *Feature-Code* [28] to categorize and classify various geographic entities. Each location is associated with a *Feature-Code*, which denotes a type of geographical location (e.g. "road", or "populated place" locations). We mapped these *Feature-Code*'s to their corresponding names to create a set  $T$ , identifying a total of 680 distinct types within GeoNames. For instance, the term "Elks Country Club" with the feature-code "S.RSRT" is mapped to its type name, "resort". The resulting  $(L, T)$  pairs were then used to create a train-test split based on  $T$ , with approximately 10% of the data allocated for testing and the remaining used for training at Task A.

For Task B, we utilized GeoNames *Feature Codes*, which are hierarchically structured to reflect varying levels of granularity in geographic features. These codes are divided into nine primary categories: "Administrative regions," "Hydrographic features," "Area," "Populated places," "Roads and railroads," "Spot features," "Terrain," "Undersea features," and "Vegetation." These categories operate at a higher level within a two-level taxonomy, resulting in 680 pairs with an "is-a" relationship. We then split the data into a 70-30 ratio to create training and test sets.

### 3.3 UMLS Ontology – Tasks A, B, and C

For generating UMLS sub-ontological sources i.e. NCI, MEDCIN, and SNOMEDCT\_US, we considered `umls-2022AB-metathesaurus-full` version of the UMLS and processed the MRCONSO files for obtaining the terms that are written in English. Next, we used the following steps for extraction of lexical terms  $L$ , their respective types  $T$ , and relations:

1. *Filtering Lexical Terms*: For each source (NCI, MEDCIN, SNOMEDCT\_US), the dataset is first filtered to extract relationships where both entities in a relationship belong to the specific source being considered. This filtering is done by matching

- the source (NCI, MEDCIN, SNOMEDCT\_US), ensuring that only triplets from that source are used. The Concept Unique Identifiers (CUIs) of these terms are then stored in a list, representing all the unique CUIs from the source.
2. *Retrieving Semantic Information:* After identifying the unique CUIs for each source, the next step is to gather semantic information about these CUIs. For each CUI, data from the MRSTY (Metathesaurus Semantic Types) file is used to obtain its Type Unique Identifiers (TUI), Semantic Type Numbers (STN), and Semantic Type Strings (STY). This information is collected and stored in a dictionary that links each CUI to its corresponding semantic types, ensuring that each TUI and STN is consistently associated with only one semantic type.
  3. *Conflict Resolution:* During the previous steps, any conflicts—where a TUI or STN might be associated with different semantic types—are checked and reported. Once the consistency of the data is verified, the final hierarchy for each source (NCI, MEDCIN, SNOMEDCT\_US) is obtained, which contains mappings from TUIs to their STNs and STYs, along with a list of all unique TUIs and STNs associated with each source, representing the hierarchical structure of entities within that specific source.

Thus, separate datasets for NCI, MEDCIN, and SNOMEDCT\_US are created, each capturing the unique semantic relationships and entity types within those sources. For Task A, we only considered CUIs and TUIs to form the task dataset. We split the datasets per source into training and testing sets with a 70-30 ratio. For Tasks B and C, since both datasets are based on the same semantic network, we leveraged this network to extract types along with their relationships. Types with 'is-a' relationships are used for Task B, while non-'is-a' relationships are used for Task C. In both cases, the datasets are split using a 70-30 ratio.

### 3.4 Gene Ontology – Tasks A, B, and C

For the Term Typing task, we needed to map lexical terms (gene products) to their generalized types, derived from three Gene Ontology (GO) sub-ontologies: Biological Process (BP), Cellular Component (CC), and Molecular Function (MF). To collect relevant annotations, we used a Python script to query the GO Lookup Service (GOLR) via the following API: <https://golr-aux.geneontology.io>. The query retrieved annotations containing information such as gene product names (bioentity name), labels (annotation class label), and the associated ontology aspect. The dataset was then grouped by the aspect field, which corresponds to the sub-ontology (BP, CC, or MF), and duplicates were removed. After gathering and preprocessing the data, we created separate datasets for each sub-ontology, organizing gene products ( $L$ ) and their corresponding types ( $T$ ). To ensure the quality of the dataset, we applied a frequency threshold of 200, filtering out low-frequency terms, thus reducing noise. Subsequently, the dataset was divided into training and test sets, with a 70-30 split to ensure a robust evaluation of models performing the Term Typing task. The resulting datasets were sufficiently large, with unique term counts for each sub-ontology, ranging from 323 to 792.

For Task B, the objective was to identify hierarchical relationships (i.e., "is-a" relations) between the generalized types from Task A. We used the GO hierarchical structure, which defines relationships as edges between nodes representing different ontology term types. Using the GO ontology file, we extracted nodes and edges from the ontology graphs and then filtered the edges to retain only those that represent "is-a" relations. We then generated pairs of term types representing the child-parent relation-

```
SELECT DISTINCT ?class ?label WHERE {
  ?class a owl:Class;
        rdfs:label ?label .
  FILTER (lang(?label) = 'en')
}
```

**Figure 2.** DBO SPARQL query for retrieving leaf classes for task A.

```
SELECT DISTINCT ?term ?label WHERE {
  ?term a <leaf_class> ;
  rdfs:label ?label .
  FILTER (lang(?label) = 'en')
}
LIMIT 100
```

**Figure 3.** DBO SPARQL query for retrieving 100 terms for given leaf class. The *leaf\_class* is a place holder for replacing it with leaf class and querying for terms.

ships (sub, obj). These pairs were split into training and test sets based on the unique term types involved, ensuring that no term appeared in both sets.

Finally, for Task C, we curated a dataset of semantic relationships between term types discovered in Task A. The relations are encoded in the GO using properties such as regulates, part of, and occurs in. We parsed the ontology to identify edges representing these relations, using a predefined set of relation mappings. Edges that matched the specified relation types were categorized into training and test sets. Similar to Task B, we ensured that there was no overlap in the relations between the training and test sets. The final dataset for Task C contained 10,538 training triplets and 7,234 test triplets, spanning multiple non-taxonomic relations.

### 3.5 DBpedia Ontology – Tasks A and B

We have used DBpedia Ontology (DBO) for both Task A and Task B, leveraging the structure and data provided by DBpedia’s SPARQL endpoint. The datasets from this ontology has been utilized in a zero-shot setting, meaning it was used exclusively for testing without any prior training. The models were evaluated directly on these unseen tasks, without exposure to any data from the specific domain during training, emphasizing their generalization capabilities for Task A and Task B.

For Task A, we queried DBpedia for leaf classes and their associated terms in English. Leaf classes were identified using the SPARQL query as described in Figure 2, which retrieves all classes with English labels. For each leaf class, we queried up to 100 terms that belong to the class, again filtering for English terms using the SPARQL query provided in Figure 3. The results of these queries were aggregated into terms and their respective types, forming the dataset for Task A.

For Task B, we queried DBpedia’s subclass ('is-a') hierarchy to generate parent-child relationships between taxonomic types. The SPARQL query, as described in Figure 4, retrieved subclass relationships where both parent and child have English labels. The resulting dataset contains hierarchical type pairs of "is-a" relations, with the taxonomic types stored as lists. This dataset serves as the input for our Taxonomy Discovery task.

```
SELECT DISTINCT ?childLabel ?parentLabel WHERE {
  ?child rdfs:subClassOf ?parent .
  ?child rdfs:label ?childLabel .
  ?parent rdfs:label ?parentLabel .
  ?child a owl:Class .
  ?parent a owl:Class .
  FILTER (lang(?childLabel) = "en")
  FILTER (lang(?parentLabel) = "en")
}
```

**Figure 4.** DBO SPARQL query for creating "is-a" relationships between taxonomic types for Task B.

```
PREFIX obo-term: <http://purl.obolibrary.org/obo/>
SELECT ?s ?label ?definition FROM <http://purl.obolibrary.org/obo/merged/FOODON> {
  ?s a owl:Class .
  ?s rdfs:label ?label .
  ?s obo-term:IAO_0000115 ?definition .
}
```

**Figure 5.** FoodOn SPARQL query for extract entity labels and definitions for Task A.

### 3.6 Food Ontology - Tasks A, B, and C

For Food ontology (FoodOn), we construct datasets for tasks A, B, and C. All tasks are designed to evaluate models in a zero-shot setting. For Task A, we queried FoodOn to retrieve leaf classes (i.e., specific entity types) and associated terms. The SPARQL query as described in Figure 5 was used to extract entity labels and definitions, ensuring that only classes with English labels were included. The output from this query was processed to assign terms to one of the predefined high-level categories such as "Food", "Environment", "Agronomy", etc. This resulted in a dataset where each term is labeled with its corresponding class type (e.g., "Food", "Plant", etc.).

For Task B on taxonomy discovery, we extracted hierarchical relationships between classes by retrieving *rdfs:subClassOf* relationships from the FoodOn. We used the SPARQL query (presented in Figure 6) to obtain parent-child pairs of classes in English, capturing the taxonomic structure. This resulted in a taxonomy dataset with pairs of parent and child concepts, which we used to evaluate how well models can uncover subclass relationships in a zero-shot context.

For the Task C, we focused on extracting object properties that represent non-taxonomic relations between entities. The Figure 7 SPARQL query was used to retrieve all object properties and their labels from the FOODON ontology. We then applied these relations to extract triples of the form (head entity, relation, tail entity), where each triple represents a non-taxonomic relationship between two entities. This yielded a dataset with various relation types and corresponding triplets, allowing us to evaluate models' performance in predicting non-taxonomic relationships.

### 3.7 Schema.org – Task B

We also leveraged the Schema.org ontology to generate a dataset for Task B, with a primary goal of extracting hierarchical relations between concepts, enabling the evaluation of how well models can identify 'is-a' relationships within a taxonomy. We ex-

```
PREFIX obo-term: <http://purl.obolibrary.org/obo/>
SELECT DISTINCT ?childLabel ?parentLabel
FROM <http://purl.obolibrary.org/obo/merged/FOODON> WHERE {
  ?child rdfs:subClassOf ?parent .
  ?child rdfs:label ?childLabel .
  ?parent rdfs:label ?parentLabel .
  ?child a owl:Class .
  ?parent a owl:Class .
  FILTER (lang(?childLabel) = "en")
  FILTER (lang(?parentLabel) = "en")
}
```

**Figure 6.** FoodOn SPARQL query to obtain parent-child pairs of classes in English for Task B.

```
FROM <http://purl.obolibrary.org/obo/merged/FOODON> WHERE {
  ?property a owl:ObjectProperty .
  ?property rdfs:label ?propertyLabel .
  FILTER (lang(?propertyLabel) = "en") .
}
ORDER BY ?propertyLabel
```

**Figure 7.** FoodOn SPARQL query to extract object properties that represent non-hierarchical relations in English for Task C.

tracted subclass relationships from the Schema.org taxonomy by processing the ontology. First, we filter out irrelevant concepts by excluding root concept `Thing` or other irrelevant RDF classes like `rdf-schema#Class`. Next, we prepare parent-child pairs by using `subTypeOf` property, where if a child had multiple parents, we split these into separate parent-child pairs. This gave us a list of hierarchical relationships, where each pair represented a child-parent relationship. Finally, to simulate a realistic few-shot scenario, we split the types into training and testing sets. Concepts that appeared in the `subTypeOf` property were divided into two sets using an 80/20 train-test split. Parent-child pairs were then assigned to the training or testing set based on the parent concepts.

## 4 Dataset Statistics

The LLMs4OL 2024 dataset is designed to support the benchmarking of ontology learning models, with a total of 19 datasets distributed across three core tasks: Task A - Term Typing, Task B - Taxonomy Discovery, and Task C - Non-Taxonomic Relation Extraction. The largest proportion of data is allocated to the Term Typing task, given its fundamental role in associating terms with predefined types, which lays the groundwork for downstream OL processes. Moreover, Taxonomy Discovery and Non-Taxonomic Relation Extraction tasks are more specialized, focusing on hierarchical and non-hierarchical relationships, respectively. This balanced yet task-specific distribution ensures that models are tested across diverse, real-world learning scenarios.

**Task A - Term Typing.** Task A datasets as described in Table 1 covers both few-shot (FS) and zero-shot (ZS) evaluation phases across multiple domains. The GeoNames (A.2 FS) is the largest dataset, with over 8 million training samples and 702 thousand testing samples, making it highly significant for large-scale geographic term

**Table 1.** LLMs4OL 2024 datasets – TASK A - TERM TYPING – domains and evaluation phases. "FS" refers to the Few-Shot testing phase dataset containing train and test sets, But "ZS" refers to the Zero-shot testing phase evaluation dataset containing only test sets.

Dataset	Domain	Train	Test	Types
A.1 (FS) - WordNet	lexicosemantics	40,559	9,470	4
A.2 (FS) - GeoNames	geographical locations	8,078,865	702,510	680
A.3 (FS) - UMLS - NCI	biomedical	96,177	24,045	125
A.3 (FS) - UMLS - MEDCIN		277,028	69,258	87
A.3 (FS) - UMLS - SNOMEDCT_US		278,374	69,594	125
A.4 (FS) - GO - Biological Process	biological	195,775	108,300	792
A.4 (FS) - GO - Cellular Component		228,460	126,485	323
A.4 (FS) - GO - Molecular Function		196,074	107,432	401
A.5 (ZS) - DBO	general knowledge	-	44,724	484
A.6 (ZS) - FoodOn	food	-	18,087	12

**Table 2.** LLMs4OL 2024 datasets – TASK B - TAXONOMY DISCOVERY – domains and evaluation phases. "FS" refers to the Few-Shot testing phase dataset containing train and test sets, But "ZS" refers to the Zero-shot testing phase evaluation dataset containing only test sets. "Size" refers to ground truth "is-a" pairs.

Dataset	Domain	Train		Test	
		Size	Types	Size	Types
B.1 (FS) - GeoNames	geographical locations	476	477	204	212
B.2 (FS) - Schema.org	web content types	1,070	2,062	364	728
B.3 (FS) - UMLS	biomedical	74	76	45	51
B.4 (FS) - GO	biological	33,703	25,372	5,753	6,621
B.5 (ZS) - DBO	general knowledge	-	-	742	762
B.6 (ZS) - FoodOn	food	-	-	30,240	25,631

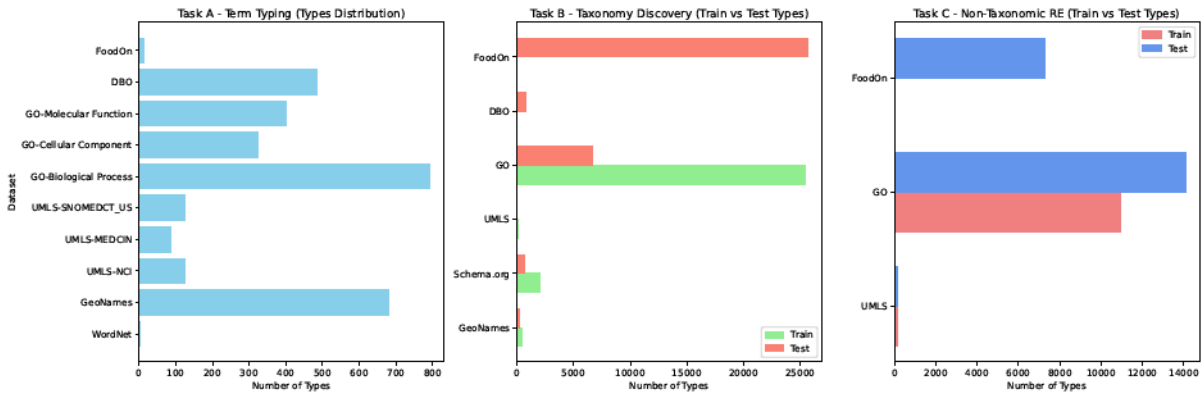
typing. Moreover, UMLS (A.3 FS) provides detailed biomedical data across three sub-ontological sources such as NCI, MEDCIN, and SNOMEDCT\_US, each with a large number of types crucial for specialized medical term categorization. The GO (A.4 FS) dataset, particularly the "Biological Process (BP)" subset, offers terms and types with the highest variety of types up to 792. DBO (A.5 ZS) and FoodOn (A.6 ZS) are important zero-shot datasets, to study the generalization of fine-tuned models.

**Task B - Taxonomy Discovery.** Task B dataset statistics are covered in Table 2, showcasing 6 datasets from different domains. The GeoNames (B.1 FS), Schema.org (B.2 FS), and UMLS (B.3 - FS) are relatively small in terms of training examples but represent unique domains (geographical locations, web content, and biomedical). Similarly, a zero-shot dataset DBO (B.5 FS) has small examples for testing which plays a real-world scenario to study the generalization of models, when they are fine-tuned. Moreover, GO (B.4 FS) stands out with over 33,703 training samples and the highest variety of types 25,372, making it key for biological taxonomy discovery. And FoodOn (B.6 ZS) is significantly large with 30,240 test samples and 25,631 types, focusing on the evaluation of the generalization of models in finding taxonomies in the food domain.

**Task C - Non-Taxonomic Relation Extraction.** Task C consists of 3 datasets, as shown in Table 3, the datasets for this task are few in comparison to task A and B datasets. The UMLS (C.1 FS), despite its moderate size, holds significance in biomedical relation extraction, focusing on multiple relation types. GO (C.2 FS) shows an imbalance in relation types, with 5 relations for training but only 2 relations for testing.

**Table 3.** LLMs4OL 2024 datasets – TASK C - NON-TAXONOMIC RELATION EXTRACTION – domains and evaluation phases. "FS" refers to the Few-Shot testing phase dataset containing train and test sets, But "ZS" refers to the Zero-shot testing phase evaluation dataset containing only test sets. "Size" refers to ground truth  $(h, r, t)$  triplets.

Dataset	Domain	Train			Test		
		Size	Types	Relations	Size	Types	Relations
C.1 (FS) - UMLS	biomedical	3,030	121	33	2,611	111	15
C.2 (FS) - GO	biological	10,538	10,901	5	7,234	14,065	2
C.3 (ZS) - FoodOn	food	-	-	-	7,086	7,298	26



**Figure 8.** LLMs4OL datasets type distributions in train and test sets.

FoodOn (C.3 ZS), with 7,086 test samples and 26 relations, highlights the complexity of non-taxonomic relations in the food domain.

**Types Distributions.** The Figure 8 highlights the complexity of the datasets across tasks. In Task A (Term Typing), the GeoNames and GO-Biological Process datasets stand out with the highest number of types, while WordNet and FoodOn have relatively fewer types, indicating simpler classification challenges. For Task B (Taxonomy Discovery), the Schema.org and GO datasets show a large number of types in both train and test phases, suggesting their complexity, while FoodOn features a high number of test types despite having no training data, making it a challenging zero-shot task. Lastly, in Task C (Non-Taxonomic Relation Extraction), the GO dataset shows a significant increase in types from train to test, and FoodOn again presents a large number of types and relations, reinforcing its difficulty in a zero-shot setting.

## 5 Conclusion

In this paper, we introduced the LLMs4OL 2024 dataset, designed to advance the field of OL by leveraging the capabilities of LLMs. The dataset encompasses three core tasks— Task A - Term Typing, Task B - Taxonomy Discovery, and Task C - Non-Taxonomic Relation Extraction—across seven distinct domains, providing a comprehensive benchmark for evaluating LLMs in diverse semantic and structural contexts. By focusing on these tasks, we aim to push the boundaries of OL and enhance the development of models capable of processing unstructured text into formalized knowledge representations. The dataset also reflects real-world challenges such as class imbalance and domain-specific variations, which are crucial for the development of robust, generalizable models. Furthermore, its integration into the LLMs4OL Challenge

at the 23rd International Semantic Web Conference (ISWC) 2024 aims to foster community engagement and encourage the exploration of novel approaches to OL.

Moving forward, this dataset and its benchmarks will provide researchers with a foundational resource to explore the intersection of LLMs and OL, promoting further innovations in knowledge extraction, classification, and relation discovery. We believe that the LLMs4OL 2024 dataset will serve as a key catalyst in the ongoing evolution of OL and its practical applications across a variety of domains.

## Data Availability Statement

The datasets supporting this article are publicly available and can be accessed via Zenodo at <https://doi.org/10.5281/zenodo.13851373>, or through the GitHub repository: <https://github.com/HamedBabaei/LLMs4OL-Challenge-ISWC2024>.

## Authors Contributions

**Hamed Babaei Giglou:** Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing - Original Draft, Writing – Review & Editing, Visualization.

**Jennifer D'Souza:** Conceptualization, Methodology, Investigation, Resources, Supervision, Project administration, Funding acquisition, Writing – Review & Editing, Visualization.

**Sameer Sadruddin:** Methodology, Resources, Data Curation.

**Sören Auer:** Conceptualization, Methodology, Review & Editing, Supervision, Project administration, Funding acquisition.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

The 1st LLMs4OL Challenge @ ISWC 2024 jointly supported by the [NFDI4DataScience initiative](#) (DFG, German Research Foundation, Grant ID: 460234259) and the [SCINEXT project](#) (BMBF, German Federal Ministry of Education and Research, Grant ID: 01IS22070).

## References

- [1] A. Maedche and S. Staab, "Ontology learning," in *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 173–190, ISBN: 978-3-540-24750-0. DOI: [10.1007/978-3-540-24750-0\\_9](https://doi.org/10.1007/978-3-540-24750-0_9). [Online]. Available: [https://doi.org/10.1007/978-3-540-24750-0\\_9](https://doi.org/10.1007/978-3-540-24750-0_9).
- [2] A. Konys, "Knowledge repository of ontology learning tools from text," *Procedia Computer Science*, vol. 159, pp. 1614–1628, 2019.
- [3] Y. Ding and S. Foo, "Ontology research and development. part 2-a review of ontology mapping and evolving," *Journal of information science*, vol. 28, no. 5, pp. 375–388, 2002.
- [4] M. Shamsfard and A. Abdollahzadeh Barforoush, "The state of the art in ontology learning: A framework for comparison," *Knowl. Eng. Rev.*, vol. 18, no. 4, pp. 293–316, Dec.



- 2003, ISSN: 0269-8889. DOI: [10.1017/S0269888903000687](https://doi.org/10.1017/S0269888903000687). [Online]. Available: <https://doi.org/10.1017/S0269888903000687>.
- [5] P. Buitelaar, P. Cimiano, and B. Magnini, *Ontology learning from text: methods, evaluation and applications*. IOS press, 2005, vol. 123.
- [6] M. Hazman, S. R. El-Beltagy, and A. Rafea, "A survey of ontology learning approaches," *International Journal of Computer Applications*, vol. 22, no. 9, pp. 36–43, 2011.
- [7] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, and H. M. Abbasi, "A survey of ontology learning techniques and applications," *Database*, vol. 2018, bay101, Oct. 2018, ISSN: 1758-0463. DOI: [10.1093/database/bay101](https://doi.org/10.1093/database/bay101). eprint: <https://academic.oup.com/database/article-pdf/doi/10.1093/database/bay101/27329264/bay101.pdf>. [Online]. Available: <https://doi.org/10.1093/database/bay101>.
- [8] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," in *The Semantic Web – ISWC 2023*, T. R. Payne, V. Presutti, G. Qi, et al., Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.
- [9] F. Petroni, T. Rocktäschel, P. Lewis, et al., *Language models as knowledge bases?* 2019. arXiv: [1909.01066](https://arxiv.org/abs/1909.01066) [cs.CL]. [Online]. Available: <https://arxiv.org/abs/1909.01066>.
- [10] B. Zhang, V. A. Carriero, K. Schreiberhuber, et al., "Ontochat: A framework for conversational ontology engineering using language models," *arXiv preprint arXiv:2403.05921*, 2024.
- [11] 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.
- [12] M. J. Saeedizade and E. Blomqvist, "Navigating ontology development with large language models," in *European Semantic Web Conference*, Springer, 2024, pp. 143–161.
- [13] R. Du, H. An, K. Wang, and W. Liu, *A short review for ontology learning: Stride to large language models trend*, 2024. arXiv: [2404.14991](https://arxiv.org/abs/2404.14991) [cs.IR]. [Online]. Available: <https://arxiv.org/abs/2404.14991>.
- [14] H. Khorashadizadeh, F. Z. Amara, M. Ezzabady, et al., *Research trends for the interplay between large language models and knowledge graphs*, 2024. arXiv: [2406.08223](https://arxiv.org/abs/2406.08223) [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2406.08223>.
- [15] 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.
- [16] A. Maedche and S. Staab, "Ontology learning for the semantic web," *IEEE Intelligent systems*, vol. 16, no. 2, pp. 72–79, 2001.
- [17] G. A. Miller, "Wordnet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [18] *Geonames geographical database*, 2023. [Online]. Available: <http://www.geonames.org/>.
- [19] O. Bodenreider, "The Unified Medical Language System (UMLS): integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, no. suppl\_1, pp. D267–D270, Jan. 2004, ISSN: 0305-1048. DOI: [10.1093/nar/gkh061](https://doi.org/10.1093/nar/gkh061). eprint: [https://academic.oup.com/nar/article-pdf/32/suppl\\_1/D267/7621558/gkh061.pdf](https://academic.oup.com/nar/article-pdf/32/suppl_1/D267/7621558/gkh061.pdf). [Online]. Available: <https://doi.org/10.1093/nar/gkh061>.
- [20] National Library of Medicine (US), *US Edition of SNOMED CT*, [http://www.nlm.nih.gov/research/umls/Snomed/us\\_edition.html](http://www.nlm.nih.gov/research/umls/Snomed/us_edition.html), Bethesda, MD, 2013.
- [21] National Cancer Institute (US), *NCI Enterprise Vocabulary Services (EVS)*, <https://www.cancer.gov/research/resources/terminology>, Bethesda, MD, 2015.

- [22] Medicomp Systems, Inc., *MEDCIN*, [http://www.medicomp.com/index\\_html.htm](http://www.medicomp.com/index_html.htm), Chantilly, VA, 2004.
- [23] S. Carbon and C. Mungall, *Gene ontology data archive*, version 2024-01-17, Zenodo, Jan. 2024. DOI: [10.5281/zenodo.10536401](https://doi.org/10.5281/zenodo.10536401). [Online]. Available: <https://doi.org/10.5281/zenodo.10536401>.
- [24] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The Semantic Web*, K. Aberer, K.-S. Choi, N. Noy, et al., Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735, ISBN: 978-3-540-76298-0.
- [25] D. M. Dooley, E. J. Griffiths, G. S. Gosal, et al., "FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration," *NPJ Science of Food*, vol. 2, p. 23, Dec. 2018. DOI: [10.1038/s41538-018-0032-6](https://doi.org/10.1038/s41538-018-0032-6). [Online]. Available: <https://www.nature.com/articles/s41538-018-0032-6>.
- [26] P. F. Patel-Schneider, "Analyzing schema.org," in *The Semantic Web – ISWC 2014*, P. Mika, T. Tudorache, A. Bernstein, et al., Eds., Cham: Springer International Publishing, 2014, pp. 261–276, ISBN: 978-3-319-11964-9.
- [27] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, *Convolutional 2d knowledge graph embeddings*, 2018. arXiv: [1707.01476 \[cs.LG\]](https://arxiv.org/abs/1707.01476). [Online]. Available: <https://arxiv.org/abs/1707.01476>.
- [28] GeoNames, *Geonames feature codes*, <https://www.geonames.org/export/codes.html>, 2024.

# silp\_nlp at LLMs4OL 2024 Tasks A, B, and C: Ontology Learning through Prompts with LLMs

Pankaj Kumar Goyal<sup>1</sup> , Sumit Singh<sup>1</sup> , and  
Uma Shanker Tiwary<sup>1</sup> 

<sup>1</sup>Indian Institute of Information Technology, Allahabad

\*Correspondence: Sumit Singh, [sumitsch@gmail.com](mailto:sumitsch@gmail.com)

**Abstract:** Our team, *silp\_nlp*, participated in the LLMs4OL Challenge at ISWC 2024, engaging in all three tasks focused on ontology generation. The tasks include predicting the type of a given term, extracting a hierarchical taxonomy between two terms, and extracting non-taxonomy relations between two terms. To accomplish these tasks, we used machine learning models such as random forest, logistic regression and generative models for the first task and generative models such as llama-3-8b-instruct, mistral 8\*7b and GPT-4o-mini for the second and third tasks. Our results showed that generative models performed better for certain domains, such as subtasks A6 and B2. However, for other domains, the prompt-based technique failed to generate promising results. Our team achieved first place in six subtasks and second place in five subtasks, demonstrating our expertise in ontology generation.

**Keywords:** Large Language Models, LLMs, Ontology Learning, Prompt-based Learning, GPT, Llama

## 1 Introduction

Ontology Learning (OL) is essential in artificial intelligence as it enables the automatic extraction and organization of knowledge from text data. Traditional methods of creating ontologies often require manual input from domain experts, resulting in a time-consuming and costly process. Recent progress in natural language processing, especially through Large Language Models (LLMs), offers a compelling alternative for automating this procedure.

The LLMs4OL paradigm, as explained in the [1], aims to use large language models (LLMs) to improve tasks in ontology learning (OL), such as term typing, discovering taxonomies, and extracting non-taxonomic relations. These models, trained on a large amount of text data, can understand complex language patterns, which can be useful for building ontologies. This study expands on this idea by using similar methods in new areas to further prove that LLMs can help automate ontology learning and reduce the need for manual input. The work in this paper extends the mentioned task to cover fifteen subtasks, as listed in table 3 and details of all tasks described in [2]. We utilized machine learning models, such as random forest and logistic regression, as

well as generative models for Task A, which involves predicting the type of a given term across multiple domains. In Task B, our objective was to find taxonomies for a given pair of terms, also across multiple domains. Task C is similar to Task B, where the goal was to find non-taxonomic relations. For Tasks B and C, we employed a prompt-based technique with generative models such as mistral 8\*7b [3], llama-3-8b-instruct [4] and gpt-4o [5] for taxonomy prediction. Our results showed that generative models performed better for certain domains, such as subtasks A6 and B2. However, for other domains, the prompt-based technique failed to generate promising results. The code of our work is available here<sup>1</sup>.

## 2 Related Work

The OL has been a major focus of research in the fields of artificial intelligence and knowledge engineering. It aims to automate the process of acquiring and structuring knowledge from text to create ontologies [6]. Traditionally, this process has relied on manual efforts by domain experts, which can be time-consuming, costly, and error-prone. To overcome these challenges, various approaches have been proposed to automate OL, primarily using lexico-syntactic pattern mining and clustering techniques [6]–[9].

One of the earliest significant works in the field of ontology learning, as highlighted by [10], involved using lexico-syntactic patterns to improve lexical ontologies like WordNet by extracting new lexicosemantic concepts and relations from large unstructured text collections. Following approaches, such as those by Hwang (2002), used iterative methods to discover types and taxonomies from text using seed terms, while [11] focused on expanding existing ontologies by reusing domain-specific ones and integrating verb patterns from text.

In recent years, developments in Ontology Learning have involved the use of machine learning techniques. Methods such as the self-organizing tree algorithm have been utilized to create hierarchical structures within ontologies. [12] introduced a TF-IDF-based term classifier and pattern finder to automatically identify domain-specific terms and relations from text. This demonstrates the evolving nature of OL methodologies.

The field of online learning has undergone significant changes with the development of Large Language Models (LLMs). These models, which are trained on extensive and diverse text collections, have shown potential in understanding complex language patterns and have been used to explore new approaches in online learning. The LLMs4OL method, introduced by [1], examines the idea that LLMs can effectively use their language modelling abilities for online learning tasks such as identifying terms, discovering taxonomies, and extracting relationships. This method demonstrates the potential of LLMs in overcoming the limitations of traditional online learning approaches, especially when customized for specific areas.

Despite the progress made, the research suggests that basic LLMs may not be skilled enough at complex ontology construction that requires deep reasoning and domain expertise. However, ongoing improvements and adjustments to LLMs for ontology learning tasks continue to demonstrate potential, providing a scalable and efficient alternative to traditional techniques.

---

<sup>1</sup><https://drive.google.com/drive/folders/1vRynlNH6Loulvcl1ymHsm6DwYKSOUoAa?usp=sharing>

### 3 Datasets

The organizers of the event have provided the dataset for each subtask. The details of the dataset can be described in [13]. Some subtasks do not have training data, and our goal is to develop zero-shot (ZS) solutions. However, training data is available for specific subtasks which require a few-shot (FS) approach. A list of all the subtask and Their statistics for all datasets of Task A and Task B are tabulated in Table 1 and Table 2, respectively. Tasks A and B are divided into various subtasks according to various domains. For example, the dataset for subtask A1 was taken from Wordnet. Also, we can see that for task A, the number of classes varies. For example, subtask A1 has four classes, and subtask A4 has 792 classes.

**Table 1.** Table shows the size of training data, testing data and number of classes for each subtask of Task A.

Task	Training Data	Testing Data	Number of classes
A.1(FS) - WordNet	40,559	9,470	4
A.2(FS) - GeoNames	8,078,865	702,510	680
A.3(FS) - UMLS(NCI)	96,177	24,045	125
A.3(FS) - UMLS(MEDCIN)	277,028	69,258	87
A.3(FS) - UMLS(SNOMEDCT_US)	278,374	69,594	125
A.4(FS) - GO(Biological Process)	195,775	108,300	792
A.4(FS) - GO(Cellular Component)	228,460	126,485	323
A.4(FS) - GO(Molecular Function)	196,074	107,432	401
A.5(ZS)	-	44,724	484
A.6(ZS)	-	18,078	12

**Table 2.** Table shows the size of training and testing data of each subtask of Task B.

Task	Training Data	Test Data
B.1(FS) - GeoNames	476	204
B.2(FS) - Schema.org	1,070	364
B.3(FS) - UMLS	74	45
B.4(FS) - GO	33,703	5,753
B.5(ZS)	-	762

## 4 Methodology

### 4.1 Methodology for Task A (Term Typing)

Term typing is a fundamental task in Natural Language Processing (NLP) that involves categorizing terms or words into predefined types or categories based on their semantic meaning, context, attributes, and relationships with other terms. The subtasks of task A involve two types: zero-shot and few-shot.

#### 4.1.1 Methodology for the few-shot subtasks (A1 to A4)

For the few-shot subtasks, we trained models using machine learning algorithms such as random forest, logistic regression, and XGBoost. Each term was converted into embeddings using the tf-idf model, where the size of each vector is equal to the total number of unique terms in the training and testing datasets.

#### 4.1.2 Methodology for the zero-shot subtasks (A5 and A6)

We have used two approaches. In the first approach, we have utilized bert [14] and sentence transformer models for the features extraction of the terms and types, and thereafter, we calculate cosine similarity between a term with all types. Most similar types are predicted as types of the term.

In the second approach, we prompted our query to the generative models. our best results for the A.6(ZS) achieved with lama-3-8b- instruct model with the following prompt:

##### Prompt:

**system\_prompt** = f"""Term typing involves Categorize terms into predefined types or categories based on their attributes. Return the answer as JSON, with each term as a key and its corresponding type from the available types as the value."""

**user\_prompt** = f"""term:{{term}},term definition:definition. classify the given term into one of types:[{{list of categories or types}}]"""

**assistant\_prompt** = """{"area of barren land": "Environment"}"""

In above prompt we have provided term and some information about term ( information about the term are extracted with the model in advance. ) with list of types and asked model to find the type of term from the given list of types. Assistant prompt is showing an example with a specific output format.

#### 4.2 Methodology for Task B (Taxonomy Discovery)

Taxonomy discovery is a task in which we need to identify the hierarchical relationship between type pairs. In this task, instances  $T_a$  and  $T_b$  are given, where  $T_a$  is the superclass (parent) of  $T_b$ , and  $T_b$  is the subclass (child) of  $T_a$ . This represents the taxonomy relationship between the two types. This task was also divided into two types of subtasks: zero-shot and few-shot subtasks.

##### 4.2.1 Methodology for the few-shot subtasks (B1, B2 and B3)

In this task, we're given training data with term types and corresponding taxonomy-related type tuples. In the testing data, only the term types are provided, and we must identify the correct taxonomic relationships from those terms. We have used few-shot prompting in multiple ways to predict the relation.

##### Few-Shot Prompting through Description-Based Approaches with GPT-4o

First, find the description of each term. Afterwards, we provided a pair of terms with descriptions, along with a list of possible relations to GPT-4o and asked to select the most suitable relation between the given terms. We have also provided some examples of pairs and their relation so that the model can understand the task with the example. To maintain efficient performance.

##### Few-shot Prompting with GPT-4o

This method is applicable for small dataset. In this approach, a list of all the terms is provided to the GPT-4o and asked to find the pairs from the given list of terms which have hierarchical relation. We have also provided some examples of pairs and their

relation so that the model can understand the task with the example. An example of prompting for the B.2(FS)-schema.org subtask with gpt-4o model is:

**Prompt:**

```
{ "role": "system", "content": "" } extract all the terms having parent child relationship means superclass subclass and return answer as a list of dict where the list contain all parent child relationship and dict contains keys as the parent and child and value of keys the parent and child which are possible from the given list of terms. return answer like this { "parent": "Animal", "child": "elephant" } } , { "role": "user", "content": f"Here is the list of terms : {test data}" }
```

**Verification-based Few-shot Prompting with mistral-22-7b**

We provided a pair of terms, along with a list of possible relations to mistral-22-7b and asked to select the most suitable relation between the given terms. Thereafter, we instruct the model to verify the relation. We have also provided some examples of pairs and their relation so that the model can understand the task with the example.

**4.3 Methodology for Task C1 (FS) UMLS (Non-Taxonomic Relationship Extraction)**

Task C is similar to Task B, except that the terms do not have a hierarchical taxonomy. We have utilized the gpt4o for prompting. For the prediction, all combinations of pairs are provided to the model and asked whether each term of a pair is related or not. We have also provided some examples of pairs and their relation so that the model can understand the task with the example.

**5 Evaluation Metric**

For Task A, the precision and f1-score are reported as the metrics for the task. We have reported the same metrics. Similarly, evaluations for Task B are reported in terms of the standard F1-score based on precision and recall.

**6 Results and Analysis**

Our best results are tabulated in Table 3 with their respective ranks. For subtasks A1, A2, A3, and A4, a comparison of results with the random forest, logistic regression, and XGboost models is shown in Table 4. The Random forest model achieved better scores, but it required more training time compared to logistic regression and XGboost.

Similarly, for subtasks A5 and A6, which are zero-shot tasks, the results with various models are shown in Table 5. The results show that GPT-4o performed better using a prompting-based approach, whereas the sentence transformer performed more effectively with a similarity-based approach.

For subtasks B and C, results with various generative models are tabulated in Table 6. The GPT-4o model demonstrated better performance for the relation extraction tasks. However, the results of B.1(FS)-GeoNames and C.1(FS)-UMLS subtasks are still challenging since no model produces good results for these subtasks.

**Table 3.** Table displays our highest F1-scores and rankings for all subtasks.

Task	F1-score	Precision	Recall	Rank
A.1(FS) - WordNet	0.90	0.90	0.90	6
A.2(FS) - GeoNames	0.44	0.75	0.31	2
A.3(FS) - UMLS(NCI)	0.69	0.87	0.57	2
A.3(FS) - UMLS(MEDCIN)	0.93	0.95	0.92	1
A.3(FS) - UMLS(SNOMEDCT_US)	0.75	0.85	0.67	2
A.4(FS) - GO(Biological Process)	0.26	0.40	0.20	1
A.4(FS) - GO(Cellular Component)	0.27	0.42	0.20	1
A.4(FS) - GO(Molecular Function)	0.29	0.41	0.23	1
A.5(ZS)	0.30	0.30	0.30	2
A.6(ZS)	0.72	0.72	0.72	2
B.1(FS) - GeoNames	0.08	0.04	0.59	3
B.2(FS) - Schema.org	0.61	0.45	0.94	1
B.3(FS) - UMLS	0.35	0.41	0.31	1
B.5(ZS)	0.21	0.14	0.42	1
C.1(FS) UMLS	0.07	0.04	0.18	1

**Table 4.** Comparison of F1-score across different machine learning models for few-shot subtasks of task A.

Task Name	Random Forest	Logistic Regression	XGboost
A.1(FS) - WordNet	0.9037	0.68	0.69
A.2(FS) - GeoNames	0.4433	0.31	0.40
A.3(FS) - UMLS(NCI)	0.6973	0.4706	-
A.3(FS) - UMLS(MEDCIN)	0.9381	-	-
A.3(FS) - UMLS(SNOMEDCT_US)	0.7552	0.7334	0.7552
A.4(FS) - GO(Cellular Component)	-	0.2725	-
A.4(FS) - GO(Biological Process)	0.24	0.2349	0.269075
A.4(FS) - GO(Molecular Function)	0.20	0.267	0.297

## 7 Conclusion

During our investigation, we explored different machine learning and generative models for ontology generation as part of the LLMs4OL Challenge @ ISWC 2024. Our approach involved using traditional machine learning models such as Random Forest, Logistic Regression, and XGBoost, as well as advanced generative models like llama-3-8b-instruct, mistral 8\*7b, and GPT-4o.

Our results showed that different approaches had varying effectiveness across tasks. For subtasks A1 through A4, Random Forest models yielded superior results, although they required longer training times compared to Logistic Regression and XGBoost. For zero-shot tasks A5 and A6, GPT-4O proved to be the most effective model, highlighting the potential of advanced generative models in scenarios where labelled data is limited. Similarly, for subtasks B and C, which focused on relation extraction, GPT-4O also outperformed other models, demonstrating its suitability for complex NLP tasks.



**Table 5.** Comparison of F1-score across different models for zero-shot subtasks of task A.

Task Name	bert-base-uncased	sentence-transformers/all-MiniLM-L6-v2	mistral 8*7b	GPT-4o-mini	llama-3-8b-instruct
A.5(ZS)	0.146	0.2001	0.2906	0.3008	-
A.6(ZS)	0.30	0.39	-	-	0.7278

**Table 6.** Comparison of the F1-scores across different models for tasks B and C.

Task Name	llama3-8b-fine-tuning-predibase	llama3-8b	gpt-4o	mistral 22*7b
B.1(FS) - GeoNames	0.083	0.041	-	-
B.2(FS) - Schema.org	-	-	0.61	-
B.3(FS) - UMLS	-	-	0.3544	0.1834
B.5(ZS)	-	-	0.2109	-
C.1(FS)-UMLS	-	0.047	0.0616	-

## Author contributions

**Pankaj Kumar Goyal:** Data curation, Methodology, Validation, Implementation.

**Sumit Singh:** Conceptualisation, Writing – Original Draft, Writing – Review & Editing, Investigation.

**Uma Shanker Tiwary:** Supervision.

## Competing interests

The authors declare that they have no competing interests.

## References

- [1] H. Babaei Giglou, J. D’Souza, and S. Auer, “Llms4ol: Large language models for ontology learning,” in *The Semantic Web – ISWC 2023*, T. R. Payne, V. Presutti, G. Qi, et al., Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.
- [2] 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.
- [3] A. Q. Jiang, A. Sablayrolles, A. Mensch, et al., *Mistral 7b*, 2023. arXiv: 2310.06825 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2310.06825>.
- [4] A. Dubey, A. Jauhri, A. Pandey, et al., *The llama 3 herd of models*, 2024. arXiv: 2407.21783 [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2407.21783>.
- [5] openai. “Gpt-4o.” (2024), [Online]. Available: <https://openai.com/index/hello-gpt-4o/>.
- [6] A. Konys, “Knowledge repository of ontology learning tools from text,” *Procedia Computer Science*, vol. 159, pp. 1614–1628, 2019, Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 23rd International Conference KES2019, ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2019.09.332>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050919315339>.

- [7] C. Fellbaum and G. Miller, "Automated discovery of wordnet relations," in *WordNet: An Electronic Lexical Database*. 1998, pp. 131–151.
- [8] C. H. Hwang, "Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information," in *Knowledge Representation Meets Databases*, 1999. [Online]. Available: <https://api.semanticscholar.org/CorpusID:11502906>.
- [9] L. Khan and F. Luo, "Ontology construction for information selection," in *14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings.*, 2002, pp. 122–127. DOI: [10.1109/TAI.2002.1180796](https://doi.org/10.1109/TAI.2002.1180796).
- [10] Z. Akkalyoncu Yilmaz, S. Wang, W. Yang, H. Zhang, and J. Lin, "Applying BERT to document retrieval with birch," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, S. Padó and R. Huang, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 19–24. DOI: [10.18653/v1/D19-3004](https://doi.org/10.18653/v1/D19-3004). [Online]. Available: <https://aclanthology.org/D19-3004>.
- [11] *OL'00: Proceedings of the First International Conference on Ontology Learning - Volume 31*, Berlin, Germany: CEUR-WS.org, 2000.
- [12] F. Dalvi, A. R. Khan, F. Alam, N. Durrani, J. Xu, and H. Sajjad, "Discovering latent concepts learned in bert," *ArXiv*, vol. abs/2205.07237, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:248810913>.
- [13] 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.
- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423). [Online]. Available: <https://aclanthology.org/N19-1423>.

# Phoenixes at LLMs4OL 2024 Tasks A, B, and C: Retrieval Augmented Generation for Ontology Learning

Mahsa Sanaei<sup>1</sup> , Fatemeh Azizi<sup>1</sup> , and Hamed Babaei Giglou<sup>2</sup> 

<sup>1</sup>University of Tabriz, Tabriz, Iran

[mahsa.san75@gmail.com](mailto:mahsa.san75@gmail.com), [fatemeazizii896@gmail.com](mailto:fatemeazizii896@gmail.com)

<sup>2</sup>TIB Leibniz Information Centre for Science and Technology, Hannover, Germany

[hamed.babaei@tib.eu](mailto:hamed.babaei@tib.eu)

\*Correspondence: Mahsa Sanaei, [mahsa.san75@gmail.com](mailto:mahsa.san75@gmail.com)

**Abstract:** Large language models (LLMs) showed great capabilities in ontology learning (OL) where they automatically extract knowledge from text. In this paper, we proposed a Retrieval Augmented Generation (RAG) formulation for three different tasks of ontology learning defined in the LLMs4OL Challenge at ISWC 2024. For task A - term typing - we considered terms as a query and encoded the query through the Query Encoder model for searching through knowledge base embedding of types embeddings obtained through Context Encoder. Next, using Zero-Shot Prompt template we asked LLM to determine what types are appropriate for a given term within the term typing task. Similarly, for Task B, we calculated the similarity matrix using an encoder-based transformer model, and by applying the similarity threshold we considered only similar pairs to query LLM to identify whatever pairs have the "is-a" relation between a given type and in a case of having the relationships which one is "parent" and which one is "child". In final, for Task C – non-taxonomic relationship extraction – we combined both approaches for Task A and B, where first using Task B formulation, child-parents are identified then using Task A, we assigned them an appropriate relationship. For the LLMs4OL challenge, we experimented with the proposed framework over 5 subtasks of Task A, all subtasks of Task B, and one subtask of Task C using Mistral-7B LLM.

**Keywords:** Large Language Models, Ontology Learning, Retrieval Augmented Generation, Term Typing, Taxonomy Discovery, Non-Taxonomic Relationship Extraction

## 1 Introduction

Ontology Learning (OL) is a critical area in knowledge representation and management, addressing the challenges of acquiring and structuring knowledge from diverse textual sources. With the rapid advancements in Natural Language Processing (NLP), particularly through the emergence of Large Language Models (LLMs), there is a compelling opportunity to enhance OL processes. LLMs have demonstrated remarkable capabilities in understanding and generating human language, making them potential candidates for automating the extraction and organization of knowledge from natural language texts. In the work of Babaei Giglou et al. [1] LLMs4OL paradigm was

introduced which investigates the hypothesis: *Can LLMs effectively leverage their language pattern recognition abilities to facilitate ontology learning?* Our approach encompasses a comprehensive evaluation of different LLM families across three primary tasks: term typing, taxonomy discovery, and extraction of non-taxonomic relationships. These tasks are evaluated using diverse ontological knowledge sources, including lexicosemantic knowledge from WordNet, geographical knowledge from GeoNames, and medical knowledge from UMLS. The empirical results from our study reveal that while foundational LLMs may struggle with the reasoning and domain expertise required for effective ontology construction, they can serve as valuable assistants when fine-tuned appropriately. This fine-tuning can alleviate the knowledge acquisition bottleneck often encountered in ontology development.

To systematically explore the capabilities of LLMs in OL, we have structured our research into three distinct tasks as described in LLMs4OL 2024 Challenge [2]:

1. **Task A – Term Typing:** This task involves classifying terms into predefined categories across various domains, such as geographical locations in GeoNames and medical terminologies in UMLS.
2. **Task B – Taxonomy Discovery:** Here, we aim to identify hierarchical relationships between term types, utilizing datasets from GeoNames and Schema.org to establish taxonomic structures.
3. **Task C – Non-Taxonomic Relationship Extraction:** This task focuses on identifying semantic relationships between terms that do not conform to hierarchical structures, with a particular emphasis on medical concepts in UMLS.

The rest of the paper is constructed as follows: In section 2 we refer to some previously conducted works. Then in section 3, we describe our methodology and after reporting the results of the study in section 4, we provide information about datasets we used in our implementations.

## 2 Related works

The construction of ontologies and knowledge graphs (KGs) has traditionally relied on human domain experts to define entities, establish relationships, and ensure data quality. However, the advent of Large Language Models (LLMs) has introduced promising avenues for automating aspects of this labor-intensive process. In the work of Kommineni et al. [3] proposed a semi-automated pipeline for constructing KGs using open-source LLMs. Their approach involves formulating competency questions (CQs), developing an ontology based on these CQs, and constructing KGs with minimal human involvement. The authors demonstrate the feasibility of their pipeline by creating a KG focused on deep learning methodologies, utilizing scholarly publications. Their findings suggest that while LLMs can significantly reduce the human effort required for KG construction, a human-in-the-loop approach remains essential for evaluating the quality of automatically generated content.

Another study [4] introduces ANGEL, a framework that integrates ontology structures and instructive prompting within LLMs for Named Entity Recognition (NER) data augmentation. This framework addresses the challenge of generating scalable training data while maintaining contextual diversity and label consistency. The experimental results indicate that ANGEL outperforms state-of-the-art methods, showcasing the potential of LLMs to enhance NER model performance, especially in low-resource scenarios. OntoChat is presented as a framework designed to facilitate conversational ontology engineering [5]. By leveraging LLMs, OntoChat supports requirement elicitation, anal-

ysis, and testing in large collaborative projects. The framework allows users to interact with a conversational agent to create user stories and extract competency questions, thus streamlining the ontology engineering process. Preliminary evaluations indicate positive feedback from domain experts, although challenges such as biases and the need for enhanced insights into implementation costs remain.

One other work presented SPIRES [6], a knowledge extraction approach that utilizes LLMs for zero-shot learning and schema-conforming query answering. SPIRES recursively interrogates prompts to extract information from input text while adhering to a user-defined knowledge schema. The method demonstrates flexibility and customization, enabling it to perform various tasks without requiring new training data. The results indicate that SPIRES can assist in knowledge curation and validation, significantly improving the efficiency of knowledge base creation. Furthermore, researchers investigate the use of LLMs to generate technical content relevant to the SAPPPhIRE model of causality. They present a method for hallucination suppression using Retrieval-Augmented Generation (RAG) to ensure the generated content is accurate and scientifically grounded. The study emphasizes the importance of the context provided to the LLM, demonstrating that different contexts can lead to varying quality in the generated responses. This research aims to build a software tool for generating SAPPPhIRE models, highlighting the potential of LLMs in technical knowledge generation [7].

In a study, L.Silva et al. [8] explore the creation of capability ontologies using LLMs. The authors conduct experiments with different prompting techniques and LLMs to generate machine-interpretable models from natural language descriptions. Their findings indicate that even complex capabilities can be accurately modeled, significantly reducing the effort and expertise required for ontology creation. The study also emphasizes the need for semi-automated quality checks to ensure the reliability of the generated ontologies. Yushi Sun and his team also investigated whether traditional knowledge graphs should be replaced by LLMs, particularly regarding their ability to capture specialized taxonomies. The authors introduce TaxoGlimpse, a benchmark for evaluating the performance of LLMs across various taxonomies. Their comprehensive experiments reveal that while LLMs perform well on common taxonomies, they struggle with specialized domains and leaf-level entities. The study suggests future research directions that combine LLMs with traditional taxonomies to create novel neural-symbolic taxonomies [9]. Recent research has started to explore the potential of LLMs in ontology matching (OM) using retrieval augmented generation (RAG), leveraging the vast amount of knowledge encoded in these models to perform more sophisticated and context-aware matching. The LLMs4OM [10] framework represents a significant advancement in this direction. It introduces an approach that employs LLMs for OM tasks through two modules dedicated to retrieval and matching, enhanced by zero-shot prompting across three ontology representations: concept, concept-parent, and concept-children. Comprehensive evaluations using 20 OM datasets from various domains demonstrate that LLMs4OM can match and even surpass the performance of traditional OM systems, particularly in complex matching scenarios using RAG.

The mentioned research collectively highlights the transformative potential of LLMs in ontology and KG construction, offering various methodologies to enhance automation and reduce the reliance on human expertise. However, they also underscore the importance of maintaining human oversight to ensure the accuracy and relevance of the generated content. As the field evolves, future research will likely continue to explore the integration of LLMs in knowledge engineering, addressing existing limitations and enhancing the effectiveness of these technologies.

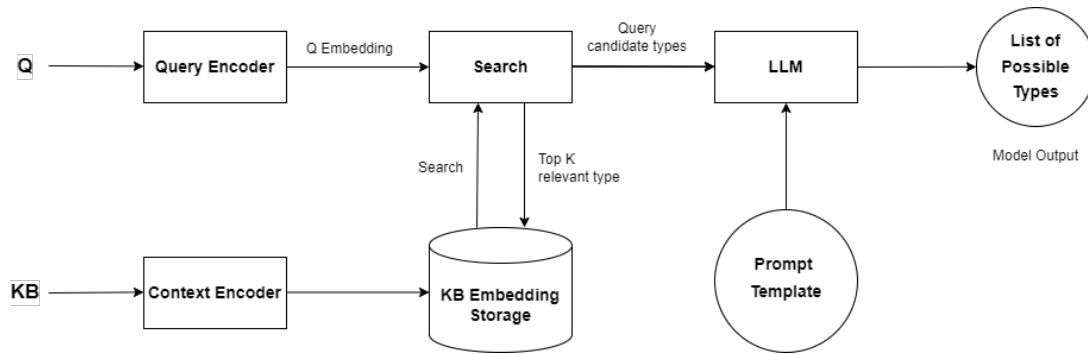


Figure 1. RAG for Term Typing Task of LLMs4OL

### 3 Methodology

#### 3.1 Task A – term typing

In Task A, the goal is to classify terms into predefined categories across various domains. We implemented a Retrieval-Augmented Generation (RAG) approach, leveraging LLMs using QLoRA approach. This setup allowed us to efficiently handle the term classification task without the need for additional fine-tuning. By integrating RAG, we aimed to enhance the accuracy and relevance of the classifications, making it suitable for a wide range of domains where terms might have clear meanings and require exact categorization.

To accomplish the task, the Figure 1 implemented to treat the types as a knowledge base (KB) and employed a Context Encoder model to generate embeddings for these types, which were then stored in the KB Embedding Storage. Specifically, we used the `dpr-ctx_encoder-single-nq-base` model [11], which is a sentence-BERT variant, to create context-aware embeddings. For any given query, we generated the corresponding embedding using a Query Encoder with `dpr-question_encoder-single-nq-base` model [11]. This dual-encoder approach facilitated a robust representation of both terms and types, ensuring that the system could effectively match terms with the most relevant types. Once the embeddings were in place, a Retrieval model searched the KB Embedding Storage to retrieve the top-k candidate types using the cosine similarity metric (we set top-k as 20). These candidate types were then passed to the LLM, specifically the `Mistral-7B-Instruct-v0.3` [12] model, which processed the candidates through a specialized prompt template (as described in Figure 2). The prompt was designed to instruct the LLM to identify the most probable types for the given term and return them in a simple Python list format, without any additional explanation. This process allowed for efficient and accurate term typing, ensuring that the most suitable types were consistently identified for each term.

#### 3.2 Task B - taxonomy discovery

In Task B: Taxonomy Discovery, the focus is on identifying "is-a" relationships between predefined types, where the goal is to determine the hierarchical child-parent relationships among these types. This process involves analyzing provided types to establish which ones serve as more general categories (parents) and which are more specific instances (children). By uncovering these relationships, we can construct or expand a taxonomy that organizes types in a structured manner, reflecting their inherent hierarchies. The overall workflow for this task is visually summarized in Figure 3.

Given a list of types as a candidate to be assigned to the term, identify the most probable types.

Return types only in the form of a Python list.  
Do not provide any explanation.

Term: <term>

Candidates: <candidates-list>

Suitable types:

Figure 2. Prompt Template for Task A - Term Typing

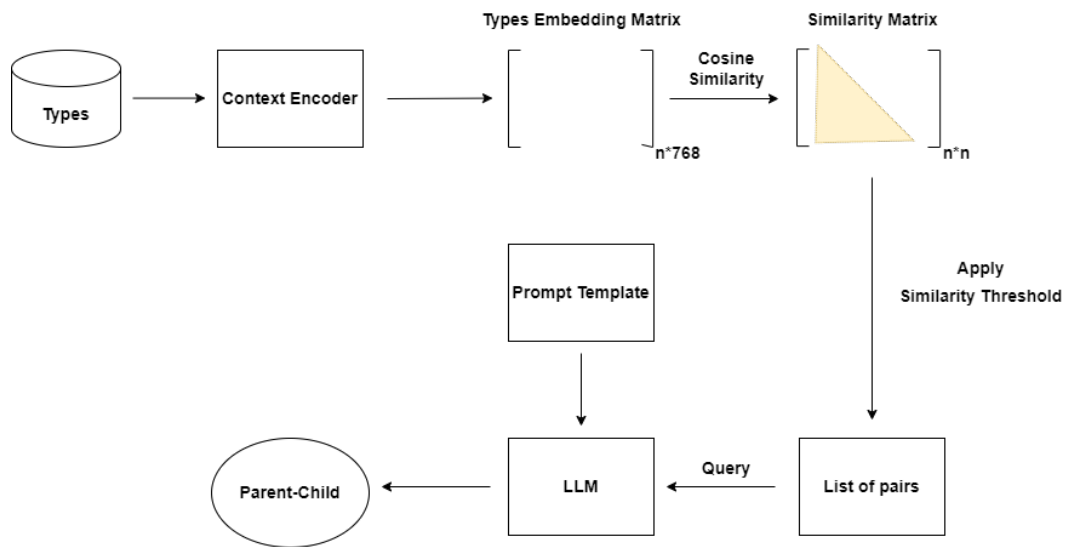


Figure 3. RAG for Taxonomy Discovery Task of LLMs4OLB

The first step in this task was to generate a types embedding matrix using a context embedding model. This matrix represents the types in a high-dimensional space, capturing their semantic similarities. To identify potential "is-a" relationships, we calculated pairwise cosine similarities between all possible pairs of types, producing a cosine-similarity matrix. This matrix serves as the foundation for detecting relationships, with each value representing how closely related two types are in terms of their embeddings. We then applied a threshold-based filter to the lower triangular part of this matrix, effectively narrowing down the list of possible type pairs that might exhibit a child-parent relationship. The filtered pairs were then passed to a Large Language Model (LLM) to assess whether a child-parent relationship exists between each pair. We designed a specific prompt template to guide the LLM in this evaluation. For each pair, the LLM was asked to determine if a hierarchical relationship was present, and if so, to identify which type is the child and which is the parent. The model was instructed to output the results in a JSON format, strictly indicating the child-parent pairs or returning an empty JSON object if no relationship was found. This structured approach ensured that the LLM's output clear and accurate taxonomy. The prompt template in Figure 4 is used in the taxonomy discover framework.

Given two types, determine whether they can have the children-parent relations or not. Then which one would be a parent and which one would be a child?

Use the following output format:

```
“  
{  
  "child": "type",  
  "parent": "type"  
}  
“
```

Notes:

- If it is not possible to establish a parent-child relationship. Just return empty '{}’.
- Do not return anything other than JSON.
- Do not provide any explanation

Term-1: <first-types>

Term-2: <second-type>

###

**Figure 4.** Prompt Template for Task B - Taxonomy Discovery

### 3.3 Task C – Non-Taxonomic Relation Extraction

In Task C: Non-Taxonomic Relation Extraction, the objective is to identify and extract triplets in the form of (head, relation, tail) from a set of given types. These triplets represent non-taxonomic relationships between types, where "head" and "tail" are types, and "relation" defines the nature of their connection. This task leverages the methodologies developed for both Task A (Term Typing) and Task B (Taxonomy Discovery), integrating them to uncover and label relationships beyond simple hierarchical structures.

The first phase of this task mirrors the approach used in Task B: we begin by identifying potential pairs of types that may have a significant relationship, treating the problem similarly to how we discovered "child-parent" relationships in Task B. We use a context embedding model to create embeddings for the types and then calculate pairwise cosine similarities to determine which pairs are closely related. By applying a threshold to the cosine similarity matrix, we filter out the most promising type pairs, which could potentially form the basis of non-taxonomic triplets.

Once the type pairs are identified, we employ an approach similar to Task A to assign the appropriate relationship (or "relation") to each pair, transforming the "child-parent" identification into a broader relation extraction. The filtered pairs are fed into an LLM using a prompt depicted in Figure 5 to determine the exact nature of the relationship between each pair. The LLM, informed by its understanding of the types, assigns a specific relation to each pair, effectively completing the triplet. This combined approach ensures we can extract meaningful and accurate (head, relation, tail) triplets, providing a comprehensive understanding of the relationships within the given set of types.



Given a head and tail type with candidate relations between them, identify the most probable relation between head and tail.

Notes:

- Return a single relation in the following format:  
{'relation': 'relation-name'}
- not provide any explanation.

Head-Type: <head-type>

Tail-Type: <tail-type>

Candidate relation between head and tail types: <candid-list>

Suitable relations:

**Figure 5.** Prompt Template for Task C - Non-Taxonomic Relation Extraction

## 4 Results

In the LLMs4OL Challenge, we participated in multiple subtasks across three major tasks: Task A (Term Typing), Task B (Taxonomy Discovery), and Task C (Non-Taxonomic Relation Extraction). Our performance was evaluated based on F1 scores, precision, and recall under both Few-Shot (FS) and Zero-Shot (ZS) testing scenario datasets of the challenge [13]. The results are presented in Table 1.

**Table 1.** Phoenixes at LLMs4OL Challenge Results Across LLMs4OL SubTasks.

SubTasks	Rank	F1	Precision	Recall
Task A - Term Typing				
SubTask A.1 (FS) - WordNet	7	0.8158	0.7689	0.8687
SubTask A.3 (FS) - NCI	5	0.0737	0.0562	0.1070
SubTask A.4 (FS) - Cellular Component	5	0.0158	0.0124	0.0217
SubTask A.4 (FS) - Biological Process	5	0.0319	0.0214	0.0622
SubTask A.4 (FS) - Molecular Function	5	0.0700	0.0485	0.1256
Task B - Taxonomy Discovery				
SubTask B.1 (FS) - GeoNames	5	0.0036	0.0019	0.0294
SubTask B.2 (FS) - Schema.org	3	0.0155	0.0079	0.3901
SubTask B.3 (FS) - UMLS	2	0.0960	0.0550	0.3778
SubTask B.4 (FS) - Gene Ontology (GO)	1	0.0164	0.0180	0.0149
SubTask B.5 (FS) - DBpedia Ontology (DPO)	2	0.0164	0.0180	0.0149
SubTask B.6 (ZS) - Food Ontology (FoodOn)	1	0.0308	0.0243	0.0420
Task C - Non-Taxonomic Relationship Extraction				
SubTask C.1 (FS) - UMLS	2	0.0273	0.0433	0.0199

Below, we provide an overview of our results and their insights.

### 4.1 Task A - Term Typing

In Task A, we participated in five subtasks focused on different ontologies and domains. Our best performance was in *SubTask A.1 (FS) - WordNet*, where we achieved an F1 score of 0.8158. This result indicates a relatively strong ability to classify terms within

the WordNet domain, with a precision of 0.7689 and a recall of 0.8687. However, our performance in the other subtasks fell short, particularly in *SubTask A.4 (FS) - Cellular Component*, where we only achieved an F1 score of 0.0158. Similar low scores were observed in *SubTask A.4 (FS) - Biological Process* (F1 = 0.0319) and *SubTask A.4 (FS) - Molecular Function* (F1 = 0.0700). These results suggest that our model struggled with more specialized biological domains, likely due to the complexity and specificity of the terms involved. Overall, the presented results show the formulation of the task with RAG is beneficial, however, fine-tuning is one of the requirements to obtain a better performance as observed in [1].

## 4.2 Task B - Taxonomy Discovery

In Task B, we explored the discovery of "is-a" relationships across various ontologies. Our best result was in *SubTask B.3 (FS) - UMLS*, where we ranked 2nd with an F1 score of 0.0960. However, the F1 scores across other subtasks, such as *SubTask B.1 (FS) - GeoNames* (F1 = 0.0036) and *SubTask B.2 (FS) - Schema.org* (F1 = 0.0155), indicate difficulties in accurately identifying taxonomic relationships in these domains. For *SubTask B.3 (FS) - UMLS* the recall score of 0.3778 shows that our approach was competitive in identifying complex relationships within the UMLS domain, however, LLM failed to find appropriate relations.

## 4.3 Task C - Non-Taxonomic Relationship Extraction

For Task C, we participated in *SubTask C.1 (FS) - UMLS*, which focused on extracting non-taxonomic relationships. Our model achieved an F1 score of 0.0273, ranking 2nd in this subtask. Despite the relatively low F1 score, this result shows that our approach was competitive in identifying complex relationships within the UMLS domain. The precision of 0.0433 and recall of 0.0199 indicate that while our model was able to correctly identify some relationships, there were challenges in capturing the full range of relevant relations, suggesting areas for further improvement.

## 5 Conclusion

In conclusion, our participation in the LLMs4OL Challenge revealed strengths in certain domains, particularly in Task A for WordNet and in Task B for Food Ontology. However, the generally low F1 scores across many subtasks highlight the challenges of term typing, taxonomy discovery, and relation extraction in highly specialized domains. These results suggest that while our approach has potential, there is significant room for improvement, particularly in enhancing the model's adaptability to diverse and complex ontologies. The implementation of this work is published in the GitHub repository for the research community at <https://github.com/MahsaSanaei/Phoenixes-LLMs4OL-ISWC>.

## Authors Contributions

**Mahsa Sanaei:** Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing - Original Draft.

**Fatemeh Azizi:** Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing - Original Draft.

**Hamed Babaei Giglou:** Conceptualization, Investigation, Review & Editing, Supervision.

## Competing interests






The authors declare that they have no competing interests.

## References

- [1] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," in *The Semantic Web – ISWC 2023*, T. R. Payne, V. Presutti, G. Qi, et al., Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.
- [2] 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.
- [3] V. K. Kommineni, B. König-Ries, and S. Samuel, *From human experts to machines: An llm supported approach to ontology and knowledge graph construction*, 2024. arXiv: 2403.08345 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2403.08345>.
- [4] Z. Luo, Y. Wang, W. Ke, R. Qi, Y. Guo, and P. Wang, "Boosting llms with ontology-aware prompt for ner data augmentation," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12 361–12 365. DOI: 10.1109/ICASSP48485.2024.10446860.
- [5] B. Zhang, V. A. Carriero, K. Schreiberhuber, et al., *Ontochat: A framework for conversational ontology engineering using language models*, 2024. arXiv: 2403.05921 [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2403.05921>.
- [6] J. H. Caufield, H. Hegde, V. Emonet, et al., *Structured prompt interrogation and recursive extraction of semantics (spires): A method for populating knowledge bases using zero-shot learning*, 2023. arXiv: 2304.02711 [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2304.02711>.
- [7] K. Bhattacharya, A. Majumder, and A. Chakrabarti, *A study on effect of reference knowledge choice in generating technical content relevant to sapphire model using large language model*, 2024. arXiv: 2407.00396 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2407.00396>.
- [8] L. M. V. da Silva, A. Köcher, F. Gehlhoff, and A. Fay, *On the use of large language models to generate capability ontologies*, 2024. arXiv: 2404.17524 [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2404.17524>.
- [9] Y. Sun, H. Xin, K. Sun, et al., *Are large language models a good replacement of taxonomies?* 2024. arXiv: 2406.11131 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2406.11131>.
- [10] H. B. Giglou, J. D'Souza, F. Engel, and S. Auer, *Llms4om: Matching ontologies with large language models*, 2024. arXiv: 2404.10317 [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2404.10317>.
- [11] V. Karpukhin, B. Oguz, S. Min, et al., "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. [Online]. Available: <https://www.aclweb.org/anthology/2020.emnlp-main.550>.
- [12] A. Q. Jiang, A. Sablayrolles, A. Mensch, et al., *Mistral 7b*, 2023. arXiv: 2310.06825 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2310.06825>.
- [13] 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.

## RWTH-DBIS at LLMs4OL 2024 Tasks A and B

### Knowledge-Enhanced Domain-Specific Continual Learning and Prompt-Tuning of Large Language Models for Ontology Learning

Yixin Peng<sup>1</sup> , Yongli Mou<sup>1</sup> , Bozhen Zhu<sup>1</sup> , Sulayman Sowe<sup>12</sup> , and Stefan Decker<sup>12</sup> 

<sup>1</sup>Chair of Computer Science 5, RWTH Aachen University, Aachen, Germany

<sup>2</sup>Fraunhofer Institute for Applied Information Technology (FIT), Sankt Augustin, Germany

\*Correspondence: Yongli Mou, [mou@dbis.rwth-aachen.de](mailto:mou@dbis.rwth-aachen.de)

**Abstract:** The increasing capabilities of Large Language Models (LLMs) have opened new opportunities for enhancing Ontology Learning (OL), a process crucial for structuring domain knowledge in a machine-readable format. This paper reports on the participation of the RWTH-DBIS team in the LLMs4OL Challenge at ISWC 2024, addressing two primary tasks: term typing and taxonomy discovery. We used LLaMA-3-8B and GPT-3.5-Turbo models to find the performance gaps between open-source and commercial LLMs. For open-source LLMs, our methods included domain-specific continual training, fine-tuning, and knowledge-enhanced prompt-tuning. These approaches were evaluated on the benchmark datasets from the challenge, i.e., GeoNames, UMLS, Schema.org, and the Gene Ontology (GO), among others. The results indicate that domain-specific continual training followed by task-specific fine-tuning enhances the performance of open-source LLMs in these tasks. However, performance gaps remain when compared to commercial LLMs. Additionally, the developed prompting strategies demonstrate substantial utility. This research highlights the potential of LLMs to automate and improve the OL process, offering insights into effective methodologies for future developments in this field.

**Keywords:** Ontology Learning, Large Language Models, Domain-specific Continual Learning, Knowledge-enhanced Prompt-tuning, Hierarchical Text Classification

## 1 Introduction

In the context of the Semantic Web community, an ontology is a formal representation of a set of concepts and the relationships between those concepts of shared conceptualizations of a domain of interest, shared by a group of people within a certain domain [1], [2]. It is often considered as a source of semantics and interoperability and used to model domain knowledge in a structured, machine-readable format. Ontology Learning (OL) refers to the process of automatic or semi-automatic creation of ontologies from text, including the extraction of terms and concepts, the extraction relationships, axiom and evaluation [3].

The recent success of OpenAI's Generative Pre-trained Transformer (GPT) has demonstrated the enormous capabilities of Large Language Models (LLMs) in natural language understanding and generation. LLMs have shown proficiency in various tasks, including machine translation, summarization, question-answering, and more recently, and now open new opportunities for enhancing OL processes, a domain-specific task. *LLMs4OL* [4] defines six key activities in OL including corpus preparation, terminology extraction, including term typing, taxonomy construction, relation extraction, and axiom discovery. The LLMs4OL Challenge ISWC 2024 [5] introduces three tasks of conceptualization (term typing, taxonomy construction, and relation extraction) and aims to explore and harness the potential of LLMs in OL within the context of the Semantic Web. It seeks to foster innovation and collaboration in the development of scalable and precise methods. This paper presents a technical report on our participation in the challenge.

## 1.1 Tasks Performed

In this study, we addressed two primary tasks defined by the LLMs4OL Challenge:

- Task A - Term Typing: Discover the generalized type for a lexical term.
- Task B - Taxonomy Discovery: Discover the taxonomic hierarchy between type pairs.

## 1.2 Main Objectives of Experiments

The primary objectives of our experiments were to assess the effectiveness of both open-source and commercial LLMs in tasks such as term typing and taxonomy discovery. Our research is centered around the following research questions, which are guiding the design of our experiments and the subsequent analysis of the results:

1. **Comparing the Performance of Commercial and Open-Source Models:** How do existing commercial models stack up against open-source models in terms of performance on the tasks defined in this competition? This question is aimed at evaluating the strengths and limitations of different models when applied to domain-specific challenges.
2. **Enhancing Open-Source Models through Training Methods:** Can the performance of open-source models be improved through the application of various training methods? This question explores whether targeted training strategies can narrow the performance gap between commercial and open-source models in these tasks.

## 2 Background and Related Work

For both tasks in the challenge, our research focuses on the following topics: Unsupervised Continual Learning and Knowledge-enhanced Prompt-tuning.

### *Unsupervised Continual Learning*

Unsupervised continual learning (UCL) focuses on the ongoing training of models using unlabeled data, a technique that has shown considerable success across various fields. In the continual pre-training scenario, models are first pre-trained on a large corpus of general text and then further pre-trained on domain-specific data to better adapt to new domains. This method has led to significant performance improvements for tasks

within new domains. For instance, Gururangan et al. [6] explored the effectiveness of domain-adaptive pre-training for NLP models. Their study demonstrated that continual pre-training on domain-specific data could significantly enhance the performance of downstream tasks, even when the amount of data is limited. Additionally, researchers such as Ke et al. [7], Scialom et al. [8], and Han et al. [9] formalized the continual pre-training scenario. They proposed pre-training strategies that effectively retain and transfer knowledge, thereby improving the generalization capabilities of the models. Their work also yielded promising results in the field of natural language processing (NLP).

The study of continual learning in large language models (LLMs) is still evolving. Wu et al. [10], wang et al. [11], and shi et al. [12] provided a comprehensive survey on continual learning for LLMs, emphasizing the necessity for regular updates to keep the models current with evolving knowledge and skills. Those studies show that continual pre-training and unsupervised continual learning offer promising avenues for enhancing the adaptability and performance of models in language domains. Therefore, we will develop our own continuous pre-training scheme based on the contextual information collected from the domain of the provided training data.

### *Knowledge-enhanced Prompt-tuning*

The integration of external knowledge into prompt-tuning methods has shown promising results in enhancing the performance of few-shot learning models. Traditional prompt-tuning approaches often struggle with tasks requiring domain-specific knowledge due to their reliance on general-purpose language models pre-trained on vast but generic datasets [13]. To address these limitations, several studies have explored the incorporation of structured and unstructured knowledge into prompt-tuning frameworks. Lu et al. [14] proposed the Medical Knowledge-enhanced Prompt Learning (MedKPL) model to improve diagnosis classification from clinical text. MedKPL leverages both structured knowledge from medical knowledge graphs and unstructured knowledge from online resources, integrating them into the prompt templates. The model demonstrated superior performance in standard and low-resource settings, showcasing its robustness and transferability across different medical departments. Similarly, liu et al. [15] introduced the Structured Knowledge Prompt Tuning (SKPT) method, which enhances prompt learning by embedding structured knowledge directly into the prompt sequences. This approach utilizes open information extraction to generate knowledge triples, which are then incorporated into the prompt templates. SKPT has shown effectiveness in few-shot text classification tasks, outperforming traditional methods on benchmark datasets.

These advancements highlight the potential of knowledge-enhanced prompt-tuning to address the challenges of domain-specific tasks in NLP. By integrating external knowledge into prompt templates, these methods not only improve model performance but also enable better generalization to unseen data and low-resource scenarios.

## **3 Methods**

In this section, we detail the methodologies employed in our approach to the LLMs4OL Challenge at ISWC 2024. Our approach depicted in Figure 1 is structured into three main stages data augmentation, model training, and inference. Each stage involves specific techniques and processes designed to optimize the performance of LLMs in term typing and taxonomy discovery.

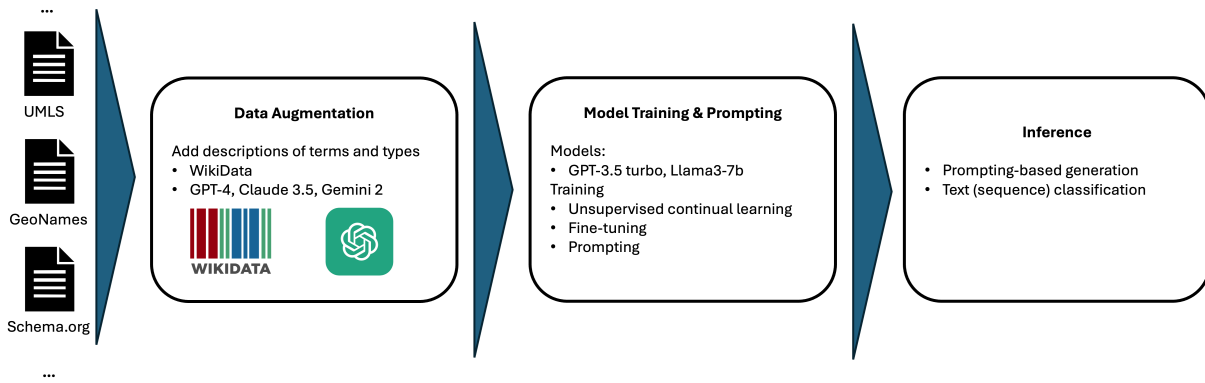


Figure 1. Overview of Research Methodology

### 3.1 Data Augmentation for Terms and Types

We participated in both Task A and Task B of the competition, each of which provided a set of ontologies [16]. Task A included the ontologies **WordNet** [17], **GeoNames** [18], **UMLS** [19], and **GO** [20]. The training data for Task A comprised three components: an optional context sentence (available only for WordNet), a lexical term, and a conceptual term type (provided as a list if the term could be assigned multiple types). In contrast, the test dataset for Task A only provided the optional context sentence (again, only for WordNet) and the lexical term.

Task B included the ontologies **GeoNames** [18], **UMLS** [19], **GO** [20], and **Schema.org** [21]. The training data for Task B was structured as pairs of types in the format  $\{T_a, T_b\}$ , where  $T_a$  represents the parent (superclass) and  $T_b$  represents the child (subclass). The test dataset for Task B provided only a series of types, requiring us to identify pairs with an "is-a" relationship among them.

During model training, we collected contextual information for terms and types using three methods described in this section, based on the data provided in the training datasets. Notably, for **GeoNames** [18], we gathered contextual information for all types and terms that appeared in both the Task A and Task B training datasets. For **Schema.org** [21], we collected information for all types present in both the training and test datasets of Task B. For **UMLS** [19] and **GO** [20], we collected contextual information only for the types found in the Task B training dataset. No contextual information was collected for **WordNet** [17].

#### 3.1.1 Data Collection from Wikipedia

We search for term or type descriptions through publicly available sources, such as Wikipedia. We utilize the Wikipedia API [22] to retrieve relevant term definitions by sending HTTP requests and parsing the returned JSON data. After obtaining the raw content, we perform data cleaning to remove irrelevant or redundant information, ensuring the data is more standardized and structured. The specific cleaning steps include:

- **Filtering Parser Instructions:** Use regular expressions to match and remove magic words such as "`__NOTOC__`" that control page content display and behavior but are not useful for our training purposes.
- **Filtering File/Image Links:** Remove file or image links starting with File, Image, or Media.
- **Filtering References and Tables:** Remove content within "ref" and "table" tags.

- **Handling Category Links:** Retain category links but remove the category prefixes.
- **Merging and Cleaning Content:** Combine various parts of the content and remove extra line breaks and whitespace characters.

This data collection and cleaning process ensures that the data obtained from Wikipedia is of high quality and consistency. The cleaned content is used for subsequent data analysis and model training. It is also suitable for collecting the definitions of various types.

### 3.1.2 Data Collection using Commercial Language Models

We employ commercial large language models with network search capabilities to collect a large amount of term-related information required for training, such as OpenAI's GPT-4o [23], Anthropic's Claude-3 [24], and Microsoft's Copilot [25]. We use zero-shot prompting to access these models through their API interfaces, and after cleaning the returned data, it is used as training data to enrich the contextual information of terms and types. Figure 2 demonstrates the example prompt we used to collect context information for the GeoNames dataset.

**Inference Prompt**

*Here is a geographical term: "{name}". Translate the given geographical term into English. Provide geographical information in plain text without any markdown formatting or reference links. Ensure that all provided information is useful for discovering implicit relationships with other geographical terms. Do not include the relationships in the result.*

**Figure 2.** Template used to prompt commercial LLMs to generate relevant information about terms in dataset GeoNames.

As an example, we present a data entry from the GeoNames training dataset for a given term: "Cascade Bluff". Using the prompt provided above, we obtained the following response from GPT-3.5-Turbo: "Cascade Bluff is a geographical location that can be translated into English. It is a place characterized by steep slopes or cliffs formed by the cascading of water. The term "Cascade" refers to the manner in which water descends rapidly and forcefully, creating a series of small waterfalls or a large waterfall. "Bluff" indicates a high, steep, or vertical cliff or bank. Geographically, Cascade Bluff is likely to be situated near a body of water, such as a river, where the cascading effect is prominent. It could be part of a hilly or mountainous region with pronounced cliffs or bluffs."

### 3.1.3 Direct Access to Ontology Data Sources

We accessed the ontology data sources associated with the datasets directly through their provided APIs or by downloading their datasets. This allowed us to retrieve and store contextual information pertinent to the training data. The contextual information refers to the explanatory text descriptions of the terms and types included in the training set, as provided within the ontology datasets. For instance, in the case of the type "lake", we extracted the following descriptive explanation from the GeoNames [18]: "a large inland body of standing water". The datasets utilized include **GeoNames** [18], **UMLS** [19], **GO** [20], and **Schema.org** [21].



## 3.2 Model Continual Learning

### 3.2.1 Domain-specific Continual Training

This approach involves domain-specific continual training of the selected open-source model. We enriched the training data using the context information for terms and types obtained through the method described in Section 3.1. For the GeoNames dataset, we collected context information for all terms and types present in the training datasets of Task A and Task B. For other datasets, we collected only the context information for types.

Figure 3 we present the training prompts constructed using type information alone and using both type and term information. These texts were used to train the model in a CausalLM manner. An analysis of the types contained in the GeoNames dataset reveals that there are 9 level-1 types and 671 level-2 types. In the prompt template provided below, "L2" represents the type of the term from the training dataset, while "L1" denotes the super-type of that type.

```
Training Prompts

# Prompt using type information alone

    Type: "{type_name}", {type_info}

# Prompt using both type and term information

    The term "{term}" from the GeoNames dataset. {term_info}. It falls under
    the top-level classification of "{L1}". Given this description, it can be logically
    inferred that "{term}" should belong to the specific sub-category "{L2}" within
    this top-level classification. "{L2}" is described as: {L2_info}. Based on this
    inference, the type of this term is determined to be "{L2}".
```

Figure 3. Template of training prompts for Method A

### 3.2.2 Fine-tuning

This method describes the fine-tuning of an open-source LLM for a specific task, aimed at addressing the downstream task of Sequence Classification. We employed different training strategies for Task A and Task B.

#### Task A

We started with the training data provided by the competition, which included terms and their corresponding types. The objective was to transform this data into a format suitable for model training. Specifically, we assigned a unique numerical label to each type and created a new dataset comprising these labels and their corresponding terms. We then fine-tuned the model using this processed data.

#### Task B

We begin with the training data provided by the challenge, which included a hierarchical parent-child relationship. First, we needed to transform these hierarchical relationships into a data format suitable for model training. Specifically, we generated textual de-

scriptions and context information for each relationship and assigned a label of 1 to these positive samples. To enable the model to distinguish between correct and incorrect hierarchical relationships, we constructed negative samples by reversing the parent and child entities and assigning a label of 0 to these negative samples. Finally, we combined the positive and negative samples to create the final training dataset. Below are data samples from the dataset for GeoNames [18]. We then fine-tuned the model using this processed data.

```
Task B Data Processing

# Unprocessed Training Data

"parent": "mountain, hill, rock",
"child": "karst area"

# Processed Prompts

"{parent} is the superclass / parent / supertype / ancestor class of {child},
They are two geographical terms. '{parent}': {parent_info} '{child}': {child_info}"
"{child} is the subclass / child / subtype / descendant class of {parent}, They are two
geographical terms. '{parent}': {parent_info} '{child}': {child_info}"
```

Figure 4. Data Processing Example for Method B - Task B

### 3.2.3 Domain-specific Continual Training Followed by Task-specific Fine-tuning

First, we performed unsupervised continual training on the specific dataset using the method described in Section 3.2.1. Next, we further fine-tuned the model for the specific task using the approach outlined in Section 3.2.2. The methods and training texts used in the phased training process were consistent with those previously described.

## 3.3 Prompting

We experimented with various prompts and ultimately selected the ones that demonstrated the best performance for the challenge. Below, we provide the templates for these prompts, each specifically designed for different tasks and models:

- 1. Prompt template for inference using GPT-3.5, applicable to all datasets in Task A:** "You are provided with a term: '{term}' from the {dataset name}, {short description about dataset}. Based on the information you have and can find on the internet, provide only the most likely type for this term. This type is strictly defined within the following list: {type\_list}. You must only give me the type. Nothing else." Here, {type\_list} represents the set of all types that appear in the training dataset for Task A's {dataset name}.
- 2. Prompt template for inference using GPT-3.5, specifically for the GeoNames dataset in Task A:** The GeoNames dataset contains 9 level-1 types and 671 level-2 types. First, GPT determines which of the 9 level-1 types the term belongs to. Then, based on the first result, GPT infers which level-2 type under the identified level-1 type the term belongs to. The prompts used for these inferences are similar to those described previously, with the key difference being the {type\_list}

provided in the prompt, which varies according to the hierarchical level being inferred.

3. **Prompt template for inference using the model trained with Method A (Section 3.2.1), specifically for the GeoNames dataset in Task A:** "The term '{term}' from the GeoNames dataset." The generated text needs to extract the type information following the keyword "sub-category" using a regular expression, and this type information is returned as the result.
4. **Prompt template for inference using the model trained with Method B (Section 3.2.2), applicable to the UMLS and GO datasets in Task A:** Use the term directly from the test data.
5. **Prompt template for inference using the model trained with Method B (Section 3.2.2), specifically for the GeoNames dataset in Task B:** "{parent} is the superclass of {child}." Here, {parent} and {child} are paired combinations from the type list provided in the test data.
6. **Prompt template for inference using the model trained with Method C (Section 3.2.2), applicable to the GeoNames and Schema datasets in Task B:** Use the prompt provided in the fifth point above.

## 4 Evaluation

In this section, we present the evaluation of our approach to term typing and taxonomy discovery as part of the LLMs4OL Challenge at ISWC 2024. The evaluation focuses on assessing the performance of our models in accurately classifying terms and constructing taxonomic hierarchies.

### 4.1 Experimental Setup

The evaluation was conducted on datasets provided for the LLMs4OL challenge, consisting of diverse sets of terms and their corresponding types and hierarchical relationships, mentioned in the previous section.

We mainly use the Hugging Face *transformers* library for the implementation, which provides robust tools for training and deploying state-of-the-art large language models, and the pre-trained model from Hugging Face model repository *meta-llama/Meta-Llama-3-8B*. The model training and inference were conducted on a high-performance server equipped with four NVIDIA H100 80GB GPUs. The duration of training depends on the size of the training data. For instance, using the GeoNames dataset from Task A, which contains 8,078,865 terms and approximately 6 GB of extracted contextual information, training on 4 GPUs for 5 epochs takes around 200 hours to complete. Training hyper-parameters are the following:

- **Learning Rate:**  $2 \times 10^{-5}$
- **Batch Size:** 16 samples per device
- **Number of Epochs:** 5 (for each mentioned training method)
- **Weight Decay:** 0.01
- **Mixed Precision Training:** Enabled (fp16)
- **Gradient Accumulation:** Accumulate gradients over 4 steps
- **Optimizer:** AdamW

The learning rate is a critical hyper-parameter that controls the step size during the update of model weights. In each back-propagation step, the model adjusts its weights according to the gradient of the loss function, with the learning rate determining the

magnitude of these adjustments. Batch size refers to the number of samples input into the model during each training iteration. The number of epochs represents the number of complete passes through the training dataset. One epoch consists of a single forward and backward pass through the entire training dataset.

Weight decay is a regularization technique designed to prevent over-fitting by constraining the magnitude of model weights. It achieves this by adding the L2 norm of the weights to the loss function, thereby penalizing excessively large weights and discouraging overly complex models. The strength of this penalty is controlled by the weight decay parameter.

Mixed precision training [26] involves using 16-bit floating-point numbers (fp16) instead of 32-bit floating-point numbers (fp32) to represent model parameters and gradients during training. This approach can significantly reduce memory usage and increase computational speed with minimal impact on the final model performance.

Gradient accumulation is a technique employed to achieve larger effective batch sizes when memory resources are limited. Specifically, instead of updating the model weights after each forward pass, the gradients are accumulated over multiple iterations (in this case, four) and then used to update the weights. This method allows for the benefits of larger batch sizes, such as more stable gradient estimates, without the need for additional memory .

The AdamW optimizer [27] is a variant of the Adam optimizer [28] that improves upon the original by decoupling weight decay from the gradient update process, thereby mitigating the bias introduced by L2 regularization. AdamW combines the advantages of momentum methods and adaptive learning rate adjustments, making it particularly well-suited for training large-scale models.

## 4.2 Results

Table 1 and Table 2 present the evaluation results for our participation in the sub-tasks of Task A and Task B, respectively. The numbers in parentheses indicate our ranking for each metric among all participating teams in the corresponding sub-task. The results in Table 1 were obtained by evaluating GPT using the prompts provided in Section 3.3 for inference. Similarly, the results in Table 2 were derived by applying the prompts from Section 3.3 for inference on the Llama-3-8B model, which was trained using the approach described in Section 3.2.3 for Task B.

**Table 1.** Results on Task A

Subtasks	F1 Score	Precision	Recall
A.1-WordNet	0.9446 (4)	0.9446 (4)	0.9446 (4)
A.2-GeoNames	0.4355 (3)	0.4355 (3)	0.4355 (2)
A.3-UMLS (NCI)	0.1691 (4)	0.1821 (4)	0.1579 (4)
A.3-UMLS (MEDCIN)	0.4566 (4)	0.4607 (3)	0.4526 (4)
A.3-UMLS (SNOMEDCT_US)	0.4747 (4)	0.4888 (3)	0.4613 (4)
A.4-GO (Biological Process)	0.0881 (3)	0.0693 (4)	0.1207 (2)
A.4-GO (Cellular Component)	0.2178 (2)	0.1846 (2)	0.2656 (1)
A.4-GO (Molecular Function)	0.1418 (2)	0.1670 (2)	0.1231 (4)
A.5-DBpedia	0.4270 (1)	0.4270 (1)	0.4270 (1)
A.6-FoodOn	0.8068 (1)	0.8068 (1)	0.8068 (1)

**Table 2.** Results on Task B Dataset

Subtasks	F1 Score	Precision	Recall
B.1-GeoNames	0.3409 (2)	0.2400 (2)	0.5882 (3)
B.2-Schema.org	0.5733 (2)	0.5475 (1)	0.6016 (2)

Next, we will provide a detailed comparison of the performance of GPT-3.5-Turbo-0125 (referred to as G) and the Llama-3-8B model, which was trained using various methods, across the UMLS, GO, Schema.org, and GeoNames datasets. The comparative results are presented as follows:

#### 4.2.1 Comparison of Training Methods for Task A

##### *GeoNames Dataset*

The Table 3 presents the performance comparison between two models, GPT-3.5-Turbo and Trained Llama-3-8B, on the GeoNames dataset in terms of F1 Score, Precision, and Recall. The Llama-3-8B model was trained using the method described in Section 3.2.1, which enriched the training data with context information for terms and types. Although the results show that GPT-3.5-Turbo outperforms trained Llama-3-8B across all metrics, considering the parameter sizes of the two models, it is evident that our training method is highly effective.

**Table 3.** Results on Task A Dataset GeoNames

Methods	F1	Precision	Recall
GPT-3.5-Turbo	0.4355	0.4355	0.4355
Llama3-8B	0.40396	0.40396	0.40396

##### *UMLS and GO Datasets*

Table 4 and Table 5 show that GPT-3.5-Turbo outperforms the Llama-3-8B model, fine-tuned using the method described in Section 3.2.2 for Task A, across all datasets in terms of F1 score, precision, and recall. The Llama-3-8B model, trained using only term textual information without providing context, performed poorly on downstream tasks. Additionally, the evaluation results for the UMLS dataset are generally better than those for the GO dataset. This discrepancy may be due to the fact that terms in the GO dataset can be assigned to a larger number of types, whereas our prompt method identifies only the most likely type.

**Table 4.** Results on Task A for Dataset UMLS. Method G for GPT-3.5-Turbo and F for Fine-tuned Llama3-8B using Method in Section 3.2.2 for Task A

Methods	NCI			MEDCIN			SNOMEDCT_US		
	F1	P	R	F1	P	R	F1	P	R
G	0.1691	0.1821	0.1579	0.4566	0.4607	0.4526	0.4747	0.4888	0.4613
F	0.0017	0.0018	0.0016	0.0001	0.0001	0.0001	0.0048	0.0050	0.0047

**Table 5.** Results on Task A for Dataset GO. Method G for GPT-3.5-Turbo and F for Fine-tuned Llama3-8B using Method in Section 3.2.2 for Task A

Methods	BP			CC			MF		
	F1	P	R	F1	P	R	F1	P	R
G	0.0881	0.0693	0.1207	0.2178	0.1846	0.2656	0.1418	0.167	0.1231
F	0.0022	0.0027	0.0018	0.0017	0.0021	0.0015	0.0014	0.0016	0.0011

#### 4.2.2 Comparison of Training Methods for Task B

##### *GeoNames and Schema.org Datasets*

The Table 6 presents the performance of the same model (Llama-3-8B) trained using two different methods. Method F refers to the fine-tuning approach described in Section 3.2.2 for Task B, which is training a model using textual descriptions and context information for each relationship extracted from the training dataset, while Method P-F refers to the training approach described in Section 3.2.3 for Task B, which is domain-specific continual training followed by task-specific fine-tuning. By comparison, it is evident that the model’s performance on this task significantly improves after training with Method P-F. The submitted results showed in Table 7 for the Schema.org dataset were also obtained after training with Method P-F. This method appears to enhance the model’s capacity to generalize across different datasets, particularly in tasks requiring a nuanced understanding of context and relationships, as evidenced by the improved F1 scores, precision, and recall on the GeoNames and Schema.org datasets. These findings underscore the importance of incorporating domain-specific knowledge and tailored training strategies to improve the performance of large language models in specialized tasks.

**Table 6.** Results for Methods F and P-F on Task B Dataset GeoNames

Methods	F1 Score	Precision	Recall
F	0.183796	0.12199	0.372549
P-F	0.3409	0.24	0.5882

**Table 7.** Results for Methods P-F on Task B Dataset Schema.org

Methods	F1 Score	Precision	Recall
P-F	0.5733	0.5475	0.6016

#### 4.3 Discussion

The results demonstrate that GPT-3.5-Turbo (G) outperforms the Llama-3-8B model fine-tuned for Task A across the UMLS, GO, and GeoNames datasets. This suggests that GPT-3.5-Turbo is more effective in handling the diverse and complex semantic structures present in these datasets. One possible reason for this superior performance is GPT-3.5’s inherent ability to leverage a broader contextual understanding, which is crucial for accurately disambiguating terms and relationships in datasets such as UMLS and GO.

In Task B, our initial submission yielded suboptimal evaluation scores. To improve performance, we re-ran the inference process and subsequently ranked the logits

scores of the outputs in descending order. Specifically, we selected the top 500 results for the GeoNames dataset and the top 400 results for the Schema.org dataset. Upon resubmission, we observed a marked improvement in the evaluation scores. This enhancement is likely due to our revised approach to handling the output data. During the inference process, we paired the types provided in the test dataset in all possible combinations, similar to constructing a matrix. However, because the ground truth labels are sparsely distributed within this matrix, the model we trained produced a relatively high number of false positives (FP) in its predictions. By refining our submission to include only the top-ranked results, we effectively reduced the number of FPs, which led to an overall improvement in the evaluation metrics. The accompanying Table 8 and Table 9) present a comparison of the evaluation results before and after implementing this top-ranking selection approach.

**Table 8.** Results for All and Top 500 Predictions on GeoNames

Methods	F1 Score	Precision	Recall
All	0.2125	0.1226	0.7941
Top	0.3409	0.24	0.5882

**Table 9.** Results for All and Top 400 Predictions On Schema.org

Methods	F1 Score	Precision	Recall
All	0.4653	0.3299	0.7890
Top	0.5733	0.5475	0.6016

Another notable observation from the results is the discrepancy in performance between the UMLS and GO datasets. GPT-3.5-Turbo achieves better evaluation metrics on the UMLS dataset compared to the GO dataset. This difference could be attributed to the nature of the datasets themselves. UMLS terms tend to have fewer associated types, making it easier for models to accurately classify them. In contrast, GO terms can be assigned to multiple types, leading to a more challenging classification task. Our prompt-based method, which identifies the most likely type, may not be fully equipped to handle the multiplicity of types in the GO dataset, resulting in lower performance metrics. This suggests that future work should explore prompt engineering techniques or model architectures that can more effectively address multi-label classification tasks.

## 5 Future Work

Building on the findings of this study, we offer the following recommendations for future research and development. These suggestions aim to enhance the performance of open-source LLMs in the two tasks addressed in this work:

### *Comprehensive Context Gathering for Task A*

For future studies of Task A, we plan to collect comprehensive contextual information for all terms and types within the UMLS and GO datasets. Leveraging this data, we intend to train open-source models using Method P-F (as detailed in Section 3.2.3) and subsequently evaluate their performance. This approach will allow us to assess the impact of enriched context on model accuracy and effectiveness.

### *Hierarchical Training of Classifiers*

We propose a hierarchical training approach for classifiers based on type hierarchies. Specifically, for datasets with multiple levels of types, such as those with 9 top-level categories and corresponding subcategories, we will first train classifiers for the top-level categories and then separately for the subcategories. The performance of this hierarchical approach will be compared to that of a single classifier trained on all types to evaluate the benefits and limitations of each method.

### *Advancing Prompting*

Enhancing prompt engineering strategies for both GPT-3.5-turbo-0125 and Llama-3-8B could lead to improved performance across a wider range of tasks. One potential strategy involves incorporating feedback loops to continuously optimize prompts based on model outputs [29]. This iterative process may include human-in-the-loop systems or automated mechanisms designed to detect and correct errors or inconsistencies in responses. By iteratively refining prompts through continuous optimization, we can enhance overall model performance, ensuring more accurate and reliable outputs.

## **6 Conclusion**

The experiments conducted with GPT-3.5-Turbo, despite not utilizing the latest commercial models such as GPT-4o, yielded promising results. In Task A, GPT-3.5-Turbo outperformed fine-tuned open-source models. However, it's important to note that these commercial models are closed-source, and the cost associated with domain-specific fine-tuning can be prohibitive.

Incorporating domain-specific information and providing a list of classification types within the prompt significantly enhances inference performance. A particularly effective strategy involves initially classifying terms into higher-level categories before refining them into more specific subcategories. This approach not only improves classification accuracy but also reduces the number of tokens required in the prompt.

Our study offers valuable insights into the performance of GPT-3.5-Turbo and Llama-3-8B on complex NLP tasks, underscoring the importance of training methodologies and the role of contextual information in boosting model accuracy. These findings carry helpfulness for the development of future models and methodologies in natural language processing. Notably, the fine-tuned open-source models demonstrated competitive performance on the GeoNames dataset, rivaling that of GPT-3.5-Turbo.

## **Supplemental Material Statement**

Our LLM prompt templates, comprehensive results, and the entire code-base are available as supplementary material on GitHub: <https://github.com/MouYongli/LLMs4OL>.

## **Author Contributions**

**Yixin Peng:** Conceptualization, Investigation, Methodology, Data Curation, Formal Analysis, Writing - Original Draft.

**Yongli Mou:** Supervision, Visualization, Project Administration, Writing - Review & Editing.

**Bozhen Zhu:** Data Curation.



**Sulayman Sowe:** Writing - Review & Editing.  
**Stefan Decker:** Funding Acquisition, Supervision.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgement

This work was supported by the German Ministry for Research and Education (BMBF) project WestAI (Grant no. 01IS22094D).

## References

- [1] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International journal of human-computer studies*, vol. 43, no. 5-6, pp. 907–928, 1995.
- [2] C. Biemann, "Ontology learning from text: A survey of methods," *Journal for Language Technology and Computational Linguistics*, vol. 20, no. 2, pp. 75–93, 2005.
- [3] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, and H. M. Abbasi, "A survey of ontology learning techniques and applications," *Database*, vol. 2018, bay101, 2018.
- [4] 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.
- [5] 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.
- [6] S. Gururangan, A. Marasović, S. Swayamdipta, *et al.*, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schlueter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 8342–8360. DOI: [10 . 18653 / v1 / 2020 . acl - main . 740](https://doi.org/10.18653/v1/2020.acl-main.740). [Online]. Available: [https : / / aclanthology.org/2020.acl-main.740](https://aclanthology.org/2020.acl-main.740).
- [7] Z. Ke, Y. Shao, H. Lin, T. Konishi, G. Kim, and B. Liu, "Continual pre-training of language models," *arXiv preprint arXiv:2302.03241*, 2023.
- [8] T. Scialom, T. Chakrabarty, and S. Muresan, "Fine-tuned language models are continual learners," *arXiv preprint arXiv:2205.12393*, 2022.
- [9] R. Han, X. Ren, and N. Peng, "Econet: Effective continual pretraining of language models for event temporal reasoning," *arXiv preprint arXiv:2012.15283*, 2020.
- [10] T. Wu, L. Luo, Y.-F. Li, S. Pan, T.-T. Vu, and G. Haffari, "Continual learning for large language models: A survey," *arXiv preprint arXiv:2402.01364*, 2024.
- [11] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [12] H. Shi, Z. Xu, H. Wang, *et al.*, "Continual learning of large language models: A comprehensive survey," *arXiv preprint arXiv:2404.16789*, 2024.
- [13] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

- [14] Y. Lu, X. Zhao, and J. Wang, "Medical knowledge-enhanced prompt learning for diagnosis classification from clinical text," in *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 2023, pp. 278–288.
- [15] J. Liu and L. Yang, "Knowledge-enhanced prompt learning for few-shot text classification," *Big Data and Cognitive Computing*, vol. 8, no. 4, p. 43, 2024.
- [16] 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.
- [17] Princeton University, *Wordnet*, Accessed: 2024-07-28, 2024. [Online]. Available: <https://wordnet.princeton.edu/>.
- [18] GeoNames, *Geonames*, Accessed: 2024-07-28, 2024. [Online]. Available: <https://www.geonames.org/export/codes.html>.
- [19] UMLS, *Unified medical language system*, Accessed: 2024-07-28, 2024. [Online]. Available: <https://www.nlm.nih.gov/research/umls/index.html>.
- [20] Gene Ontology, *Gene ontology*, Accessed: 2024-07-28, 2024. [Online]. Available: <https://www.geneontology.org/>.
- [21] Schema.org, *Schema.org*, Accessed: 2024-07-28, 2024. [Online]. Available: <https://schema.org/>.
- [22] MediaWiki API, *Api: Main page - mediawiki*, Accessed: 2024-07-28, 2024. [Online]. Available: [https://www.mediawiki.org/wiki/API:Main\\_page](https://www.mediawiki.org/wiki/API:Main_page).
- [23] OpenAI, *Gpt-4o*, Accessed: 2024-07-28, 2024. [Online]. Available: <https://openai.com/index/hello-gpt-4o/>.
- [24] Anthropic, *Claude 3.5*, Accessed: 2024-07-28, 2024. [Online]. Available: <https://www.anthropic.com/claude>.
- [25] Microsoft, *Copilot: Ai-powered assistance in bing*, Accessed: 2024-07-28, 2024. [Online]. Available: <https://copilot.microsoft.com/>.
- [26] P. Micikevicius, S. Narang, J. Alben, *et al.*, "Mixed precision training," *arXiv preprint arXiv:1710.03740*, 2017.
- [27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
- [28] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [29] A. Madaan, N. Tandon, P. Gupta, *et al.*, "Self-refine: Iterative refinement with self-feedback," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

# TSOTSALearning at LLMs4OL Tasks A and B : Combining rules to Large Language Model for Ontology learning

Carick Appolinaire Atezong Ymele <sup>1</sup> and Azanzi Jiomekong <sup>1,2</sup>

<sup>1</sup>Department of Computer Science, University of Yaounde I, Yaounde, Cameroon

<sup>2</sup>TIB – Leibniz Information Centre for Science and Technology, Hannover, Germany

\*Correspondence: Carick Atezong, [carick.atezong@facsciences-uy1.cm](mailto:carick.atezong@facsciences-uy1.cm)

**Abstract:** This paper presents our contribution to the Large Language Model For Ontology Learning (LLMs4OL) challenge hosted by ISWC conference. The challenge involves extracting and classifying various ontological components from multiple datasets. The organizers of the challenge provided us with the train set and the test set. Our goal consists of determining in which conditions foundation models such as BERT can be used for ontologies learning. To achieve this goal, we conducted a series of experiments on various datasets. Initially, GPT-4 was tested on the wordnet dataset, achieving an F1-score of 0.9264. Subsequently, we performed additional experiments on the same dataset using BERT. These experiments demonstrated that by combining BERT with rule-based methods, we achieved an F1-score of 0.9938, surpassing GPT-4 and securing the first place for term typing on the Wordnet dataset.

**Keywords:** Ontology Learning, Large Language Models, Rules, BERT

## 1 Introduction

Knowledge acquisition from scratch is costly in time and resources. Ontology learning aims to reduce this cost. Ontology learning is the extraction of ontological knowledge from unstructured, semi-structured or fully structured knowledge sources in order to build an ontology from them with little human intervention [1].

A lot of work has been done on the extraction of ontological knowledge from several data sources such as texts [2], databases [3], XML files [4], vocabularies [5], etc and several domain such as food information [6], food composition knowledge from scientific literature [7], healthcare [8]. These works resulted into symbolic based techniques, statistical based techniques, and multi-strategy based techniques. Given that Large Language Models (LLMs) have shown significant advancements in natural language processing, Babaei et al. [9] proposed a Large Language Models for Ontology Learning (LLMs4OL) approach. The authors evaluated nine LLMs families on several datasets. These evaluations shows that foundational LLMs are not sufficiently suitable for ontology learning. However, in many context students, researchers, etc. do not always have enough resources to run LLMs such as LLaMA-7B or GPT-3.

The main goal of this study is to reply to the following research question: *"In which conditions foundations models can be used for ontology learning"*. To reply to this question, we participate to LLMs4OL 2024 challenge [10]. This challenge aims to explore the intersection of LLMs and OL. The organizers of this challenge provided train and test datasets. The GPT-4 model was run and evaluate on four of the dataset. Thereafter, the BERT-Base uncased model was chosen and a set of experimentation was conducted. These experimentation's show that by merging the strengths of LLMs such as BERT with symbolic techniques such as rules, the model obtained can be as powerfully as GPT-4.

Before presenting the methodology in Section 2.2, we present the challenge in Section 2.1, followed by the evaluation in Paragraph 2.1, the approach we used in Section 2.2 with the results in Section 3. Finally, Section 4 provides the conclusion. To facilitate the reproducibility of the results, the codes used in this study are available on our GitHub repository at <https://github.com/sudo-001/LLMs4OL-2024>.

## 2 An Approach Combining LLMs with Rules for Ontology Learning

Taking advantage of our experience in the field of ontology learning using symbolic approaches such as rules and LLMs such as BERT, we defined an approach combining LLMs and rules for ontology learning. This methodology was applied on the datasets provided by LLMs4OL challenge. Before we present this methodology in Section 2.2, the main ontology's components will be presented in Section 2.1.

### 2.1 LLMs4OL Challenge

LLMs4OL challenge aims for exploring the intersection of LLMs and OL. The following tasks were proposed by the organizers of this challenge:

- **Task A - Term Typing:** aims to discover the generalized type for a lexical term. This correspond to a concept or a class and aims to represent a category of object;
- **Task B - Taxonomy Discovery:** aims to discover the taxonomic hierarchy between type pairs;
- **Task C - Non-Taxonomic Relationship Extraction:** aims to identify non-taxonomic relation between types.

#### *Evaluation*

The organizers provided us for each dataset the train and the test dataset. To evaluate our system, we trained the model on the train data and we evaluated on the test data on the codalab platform. The evaluation was done using the Precision, Recall and F1-score.

## 2.2 Methodology

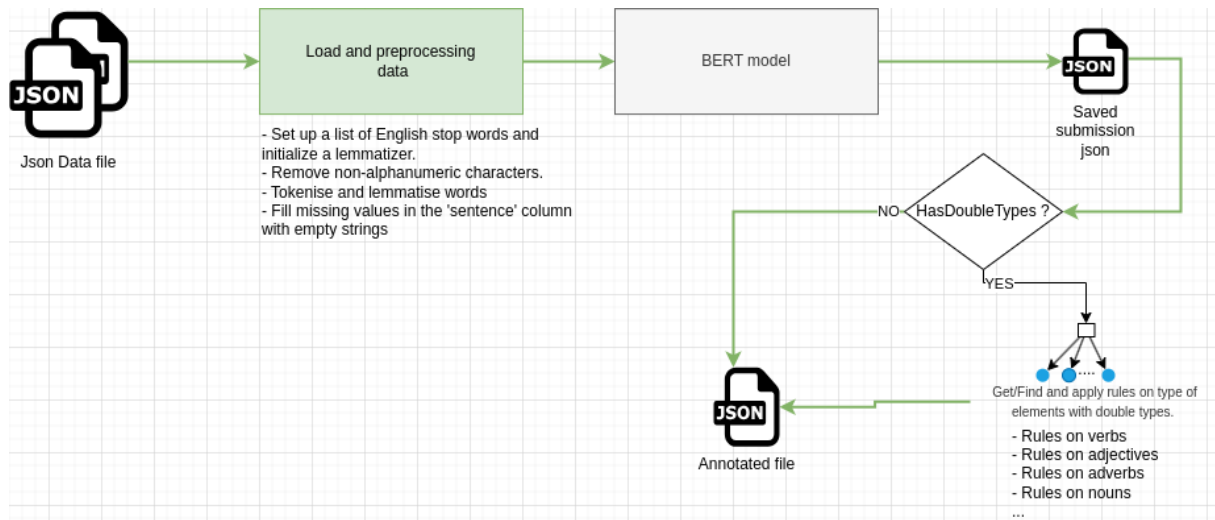


Figure 1. An Approach Combining LLMs with Rules for Ontology Learning

To enhance the process of ontology learning, we propose in this work a methodology (see Figure. 1) based on the combination of LLMs with rules derived from an in-depth analysis of the training data. This analysis involved identifying recurring patterns and contextual associations between terms and their corresponding types. This combination is described by the equations below (eq.1 and eq.2).

$$M = \{LLM_p, R_s\} \quad (1)$$

$$R_s = \{r_1, r_2, \dots, r_n\} \quad (2)$$

- $M$  : Represent our methodology.
- $LLM_p$ : this is the pre-trained LLM on the trained dataset.
- $R_s$ : this is the set of rules that characterises the dataset. An example of a rule found in the WordNet dataset is "if a term ends with 'ly' and the predicted model predict two types or more then the type is 'adverb'".

The workflow consists of the following key steps:

1. **Data Preprocessing:** This step consists of refining the dataset so as to assure that it is clean and properly structured. To this end, non-alphanumerical characters such as (!, #, -, \_,... ) are removed, text are converted into lowercase and tokenize, each token are reduce to it basal value, and the lemmatised token are combined to form the preprocessed sentence.
2. **Finetuning LLM for OL:** During this step we fine-tuned a pre-trained BERT model for our specific text classification task. Below are the additional details regarding the fine-tuning process : - **Model Choice:** We used the pre-trained BERT model (bert-base uncased) for its proven ability to capture rich contextual representations of text; - **Data preparation:** The data was pre-processed and encoded using the BERT tokenizer; - **Fine-tuning configuration:** we configured the fine-tuning with 3 epochs, a batch size of 16 for training, a warm-up steps of 500, a weight decay of 0.01.

3. **Evaluating the fine-tuned model:** In this step, the fine-tuned model is run on the test data. We do not use prompts or additional query formulations during this process. Instead, The evaluation was carried out by feeding the pre-processed test data directly into the model. The model results were then used to generate predictions for each test instance.;
4. **Evaluating the LLM output:** This step consists of evaluating the output of the test data using the precision, recall and F1-score. If the score is sufficiently high, one can stop the process. In our case we used the codalab platform to evaluate our results.
5. **Assessing the output:** This step consists of identifying the elements that are not well predicted;
6. **Complete the model with rules:** This step consists for each element identified in step 5 to defined a rule that allow us to predict the right output.

## 2.3 Experimentation environment

To evaluate the different systems for ontology learning, the organizers of the LLMs4OL provided several datasets [11]. The Table. 1 present a detailed description of the datasets for term typing and Table. 2 present the different datasets for taxonomy discovery.

**Table 1.** Overview of the datasets used in this work for task A : Term typing

Dataset	Train Size	Test Size	Number of Types
WordNet	40,559	9,470	4
GeoNames	8,078,865	702,510	680
GO-Biological Process	195,775	108,300	792
GO-Cellular Component	228,460	126,485	323
GO-Molecular Function	196,074	107,432	401

**Table 2.** Overview of the datasets used for Task B : Taxonomy Discovery task

Dataset	Train Size	Test Size
GeoNames	476	204
Schema.org	1,070	364
UMLS	74	45
GO	33,703	5,753

1. **Wordnet:** See table 1. The WordNet dataset is a large lexical database, where words are in english and organized into sets of synonyms called synsets. This dataset contains two types of entries: (1) Entry with term or group of terms accompanies with it's usage. For instance, "cover" as a term and "cover her face with a handkerchief" as the contextual sentence or the usage example. (2) Entry with terms or group of terms without example of usage. For this dataset, the task was to predict the type of terms (corresponding to Task A of the challenge).
2. **Geonames:** See table 1. GeoNames is a geographical database that contains over 8 million placenames and corresponding geographical information. It includes information such as location coordinates, population, and administrative divisions. Such as "Pic des Langounelles" a term or an entity with the type "peak". This dataset contains terms without context or usage sentence. This dataset was used for tasks A and B Taxonomy discovery.

3. **Gene Ontology (GO):** The Gene Ontology dataset (see Table 1) provides a structured vocabulary for representing gene product attributes across species. This dataset includes three domains: **Biological Process**, **Molecular Function**, and **Cellular Component**. As WordNet and GeoName, this dataset contains terms with one or multiple words. An example is following: The term "Tetratricopeptide repeat protein 19, mitochondrial" with the type "mitotic cytokinesis" for biological process.

### 2.3.1 Hardware and software

The experimentation was conducted in a controlled environment to ensure the reproducibility and reliability of our results.

- The hardware used for our experiments was a laptop Dell Precision 5510, with an Intel Core i7-6820HQ CPU running at 2.70GHz with 8 cores, 16.0 GiB of RAM, and a disk capacity of 756.2 GB.
- The operating system was Ubuntu 22.04.4 LTS.

The BERT-Base uncased was chosen as the LLM to use. In addition, we have chosen GPT-4 as a very large LLM and our goal was to determine in which conditions the foundation model can beats an LLM such as GPT-4.

### 2.3.2 Experimentation processing

The first step of the experimentation consists of evaluating the performance of GPT-4 on the test data. Thereafter, we have chosen to use BERT-Base uncased as the foundation model. Once the pre-trained model is run on the test data, a manual assessment allow us to define the set of rules to combine with the pre-trained model and the model is tested once. For instance, a manual assessment of the WordNet dataset allowed us to realize that the terms without context was the one that was not well predicted. Thus, we defined a set of rules that we applied on verb, adjectives, and adverbs.

## 3 Results and Discussion

This section presents the results of the application of our methodology (see Section 2) for the term typing (see Section 3.1) and taxonomy discovery (see Section 3.2) on WordNet, GeoName, and GO datasets.

### 3.1 Term Typing Task

The following paragraphs presents the results (accompanied with ablation study) on WordNet and Geoname datasets.

#### 3.1.1 Term Typing on WordNet Dataset

Concerning the WordNet Dataset, the BERT-Base uncased model [12] was combined with several rules obtained by assessing the dataset manually. Actually, the manual assessment allowed us to realize that when the context is not provided, BERT failed to identify the type. This allowed us to adapt the equations 1 and 2 in section 2 to the WordNet dataset and obtain the equation below.

$$R_s = \{verb_{rule}, adjective_{rule}, adverb_{rule}\} \quad (3)$$

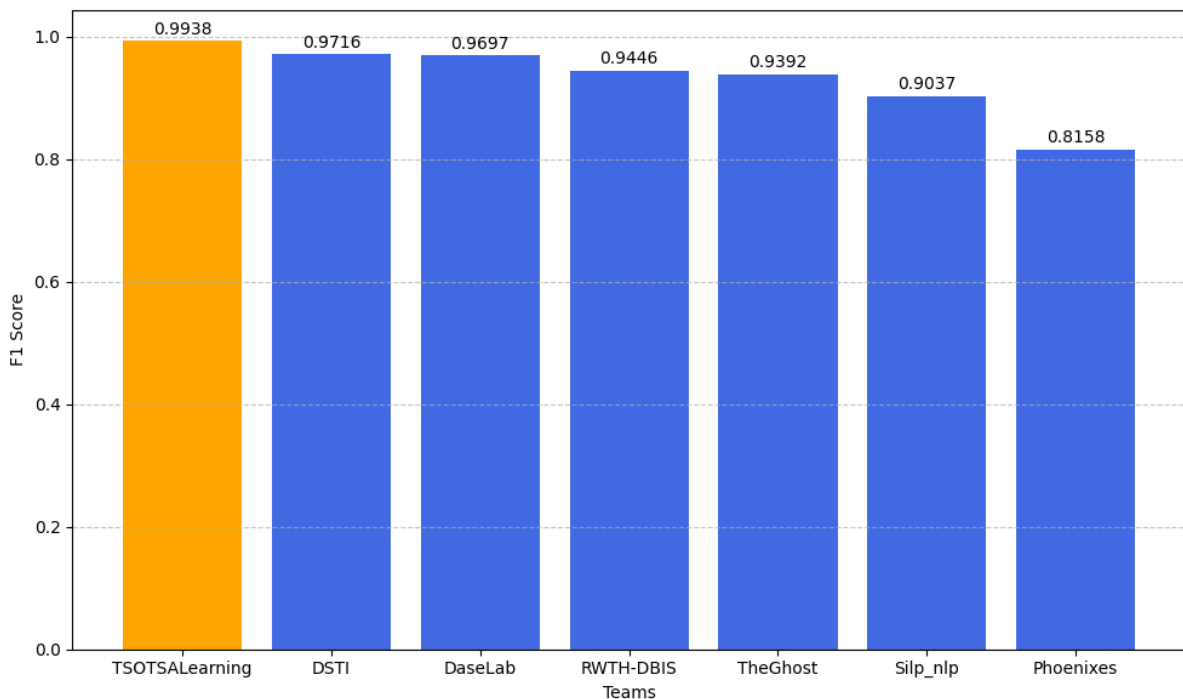
The following equations describe the rules defined in equation 3.

$$verb\_rule = \{ verb \text{ if } term \in \{ate, ify, ize\} \wedge |obj\_type| = 2\} \quad (4)$$

$$adjective\_rule = \{ adjective \text{ if } term \in \{ible, able, al, ic, ous, ful, ive\} \wedge |obj\_type| = 2\} \quad (5)$$

$$adverb\_rule = \{ adverb \text{ if } term \text{ ends with "ly"} \wedge |obj\_type| = 2\} \quad (6)$$

This model was applied on the test data provided by the organizers. Figure. 2 presents the results obtained in comparison with the results of other systems. This figure shows that the system obtained using this model is the best system. It should be noted that this system was run on a simple laptop.



**Figure 2.** Comparing the different score obtained per systems submitted to the challenge

### Ablation study

To study the impact of rules on the whole system, several parts of the rules were removed. The table 3 presents the results obtained from our approach, compared to use achieved using the th obtained using the GPTGPT-4 model and -4 model and usithe rule-based method.ng different rules. This table shows that the performance of the model depends on the completeness of the rules identified.

The low performance compared to the performance when combining BERT-Base uncased with rules, suggest that rules can be an important component when learning ontology using LLMs.

In conclusion, when BERT-Base uncased is enhances with rules for ontology learning, the model obtained can be as powerful as the one obtained using LLMs such as GPT-4.



**Table 3.** Results of the ablation study. (1)  $BERT_{bu}$ : BERT-Base uncased

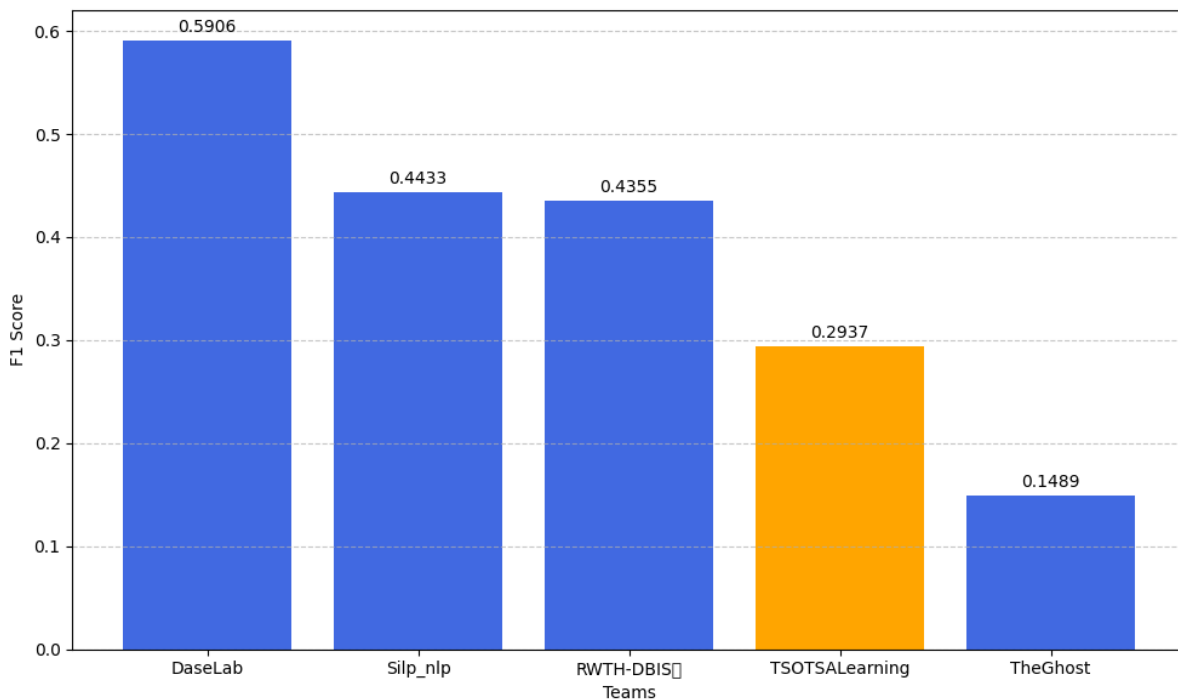
Method	Precision	Recall	F-score
$BERT_{bu}$	0.5994	0.9866	0.7457
GPT-4	0.9264	0.9264	0.9264
$BERT_{bu}$ + Verbs	0.9403	0.9403	0.9403
$BERT_{bu}$ + Adjectives	0.9332	0.9332	0.9332
$BERT_{bu}$ + Adverbs	0.9332	0.9332	0.9332
$BERT_{bu}$ + All Rules	<b>0.9938</b>	<b>0.9938</b>	<b>0.9938</b>

Given the results obtained after the experimentation on the WordNet dataset, we decided to adopt this approach for the other datasets. However, the GeoName and GO training datasets were too large and the time to finetune the model, test on the test data was not enough. It requires at least 6 days for all molecular on our training environment (see Section 2.3.1) and at least 15 days for all geonames. We were able to finetuned the BERT-Base uncased model on only **16.67%** of data for GeoName, **16.67%** of data for Cellular, **16.67%** of data for molecular. During this process, a manual assessment of the dataset allowed us to identify several rules that can be used to enhance the LLM once finetuned.

### 3.1.2 Term Typing on GeoNames Dataset

The equation 1 presents the model used for term typing on GeoNames dataset.

This model was applied to the test data provided by the organizers. Figure. 3 presents the results obtained in comparison with the results of other systems. This figure shows that the system obtained using this model has the fourth position.



**Figure 3.** Comparison of the F1-score of the TSOTSA Learning system with other systems on the GeoNames dataset

### Ablation study

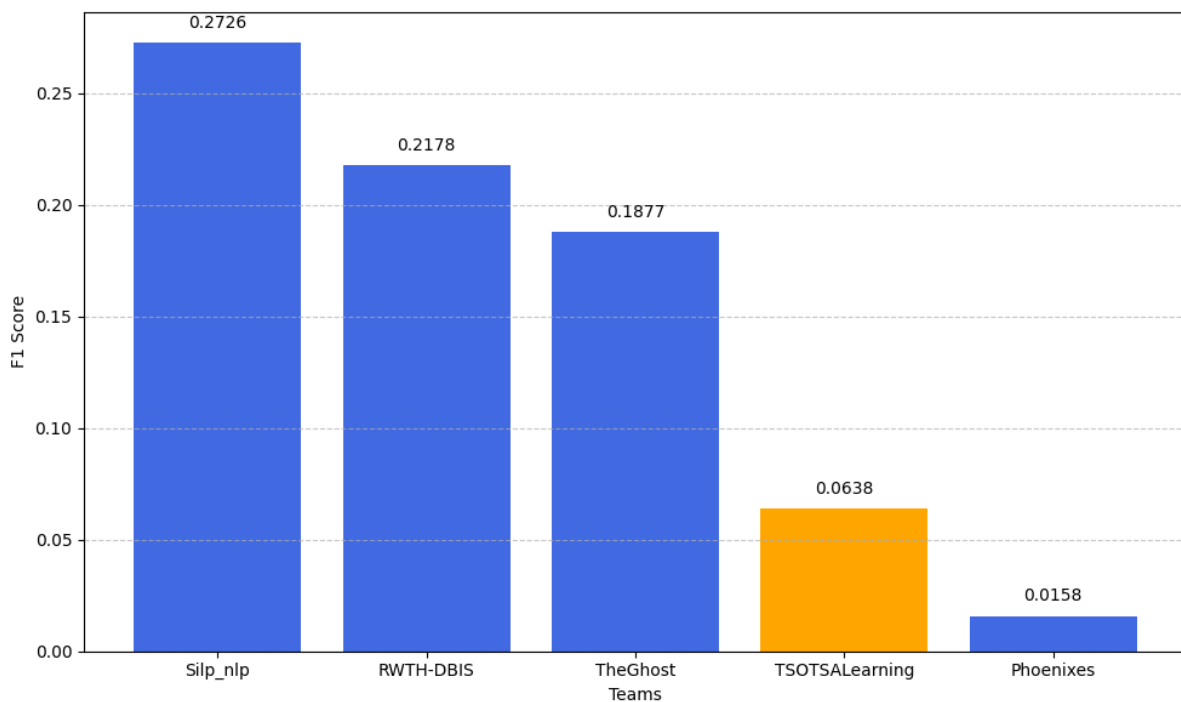
To study the impact of rules on the whole system, the rules applied on the GeoName dataset was removed and the system was evaluated on the test data. The results obtained are presented by table 4. This table shows that only rules allow to obtained the 0.2937 of F1-score. It should be noted that the model was finetuned on only **16.67%** of the training dataset.

**Table 4.** Results of the ablation study. (1) BERT\_bu: BERT-Base uncased

	rules applied	BERT_bu	GPT-4
Precision	<b>0.2937</b>	0.0000	0.0000
Recall	<b>0.2937</b>	0.0000	0.0000
F-score	<b>0.2937</b>	0.0000	0.0000

### 3.1.3 Term Typing on Cellular Component Dataset

Similar to the WordNet and the GeoName dataset, the model defined (see equation 1) was applied on the "Cellular Component Dataset". The results obtained, compared with other systems are presented by the Figure. 4.



**Figure 4.** Application of the TSOTSALearning system on test set of the Cellular dataset

### 3.1.4 Term Typing on Biological Process Dataset

Concerning the Biological Process, the BERT-Base uncased model was pretrained, combined with rules (see Figure. 5) applied to the test data and submitted on the codalab platform for evaluation.

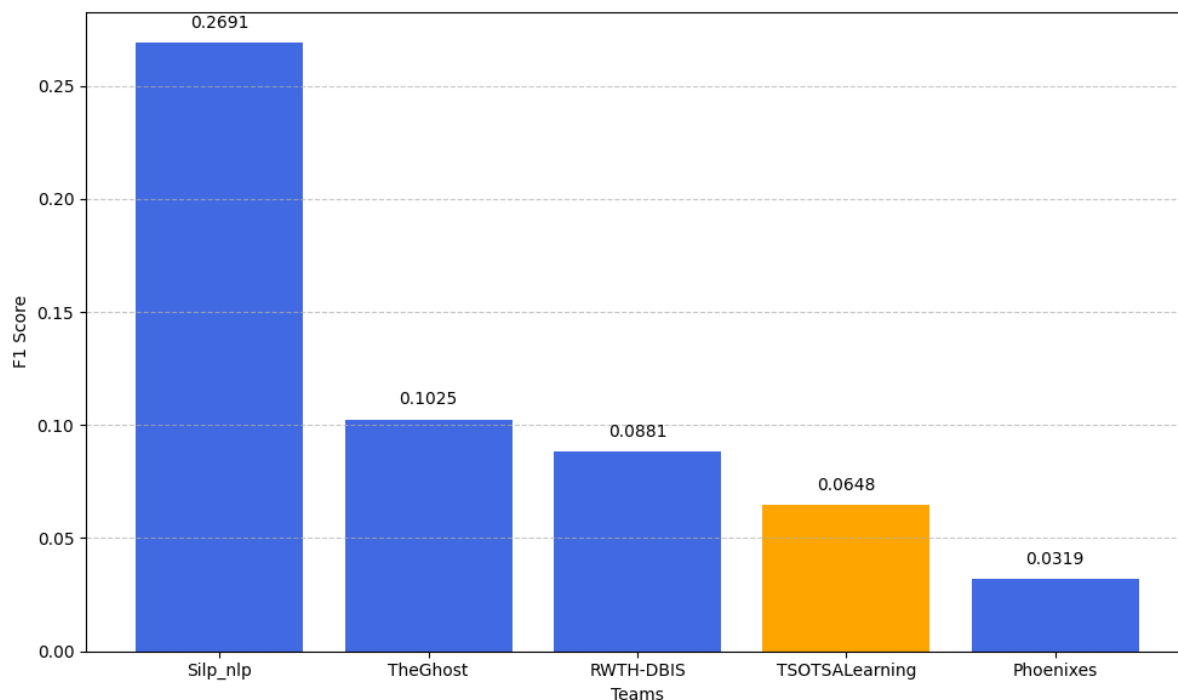


Figure 5. Application of the BERT-Base uncased model on the Biological Process dataset

### 3.1.5 Term Typing on Molecular Function Dataset

Concerning the Molecular Function, the BERT-Base uncased model was pre-trained, combined with rules and applied to the test data. Figure. 6 presents the results obtained compared to the results of other systems.

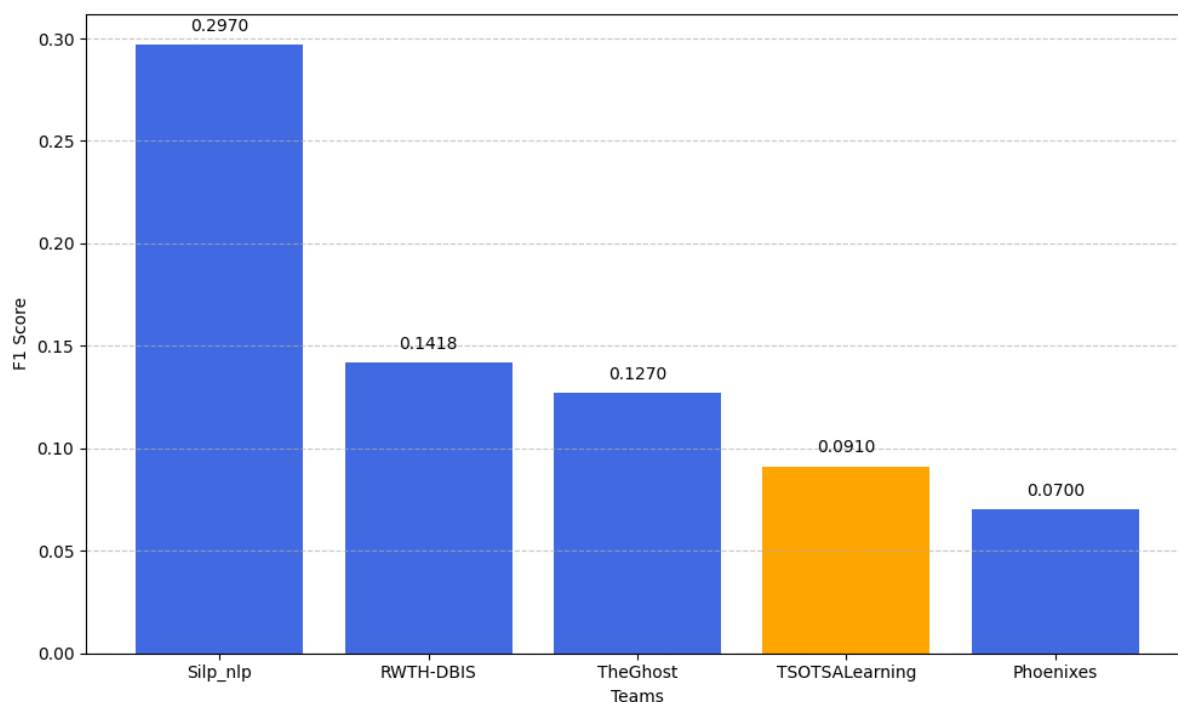


Figure 6. Application of the BERT-Base uncased model on the Molecular dataset

### 3.2 Taxonomy Discovery on GeoNames Dataset

During the taxonomy discovery task, given the time for submitting our results, only the BERT-Base uncased model was used on the GeoName dataset. Figure. 7 presents the results obtained compared to the results of other participants. This figure shows that the system proposed occupy the fourth position.

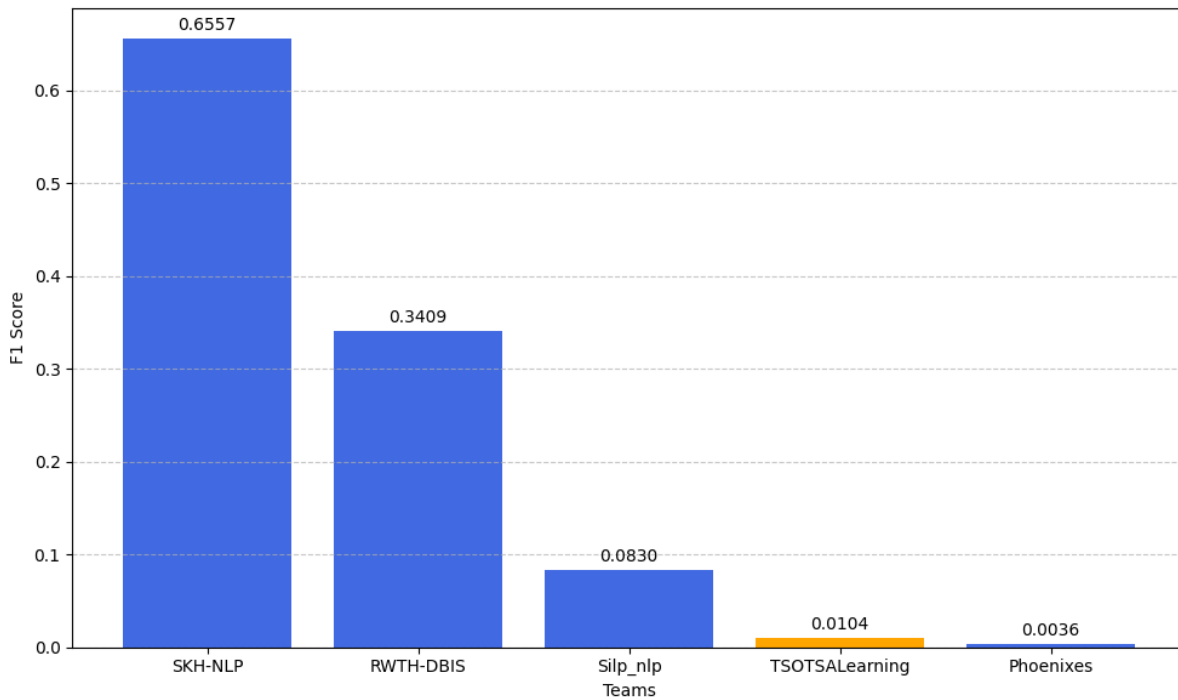


Figure 7. Application of the BERT-Base uncased model on the GeoName dataset

### 3.3 Conclusion

The results for the taxonomy discovery task, Fig 7 reveal considerable challenges, particularly relating to the accuracy of predictions. The low f1 score on Geonames, despite higher recall suggests that the model identifies many potentially relevant terms but has difficulty avoiding false positives. This highlights the complexity of taxonomic relationships and the importance of improving the accuracy of the model.

## 4 Conclusion

This research aims to determine in which conditions foundations models such as BERT can be used for ontology learning. A set of experimentation's was conducted using BERT and compared the results obtained to the results obtained using GPT-4. The results obtained on the WordNet dataset show that merging the strengths of LLMs with **rule-based strategies**, enhances the accuracy of ontology learning. The ablation study consists of comparing the performance of the LLM alone and the combination of the LLM with rules. This suggests that rules can be an important component when learning ontologies using LLMs. It should be noted that identifying rules to used is not an easy task. Future work consists of automatic detection of rules and the possibility to inject the rules in the LLM.

## Author Contributions

**Carick Appolinaire Atezong Ymele:** Conceptualization, Methodology, Software, Writing - Original Draft, Writing - Review & Editing.

**Azanzi Jiomekong:** Conceptualization, Methodology, Validation, Writing - Original Draft, Writing - Review & Editing, Supervision.

## Competing interests




The authors declare that they have no competing interests.

## References

- [1] F. J. Azanzi, G. Camara, and M. Tchuente, "Extracting ontological knowledge from java source code using hidden markov models," *Open Computer Science*, vol. 9, no. 2, pp. 181–199, Aug. 2019. DOI: [10.1515/comp-2019-0013](https://doi.org/10.1515/comp-2019-0013).
- [2] H. Zaragoza, P. M. D. Cabeza, and J. R. Sanz, "Learning ontologies from text: A survey of approaches and techniques," *Journal of Computer Science and Technology*, vol. 32, no. 1, pp. 1–14, 2017. DOI: [10.1007/s11390-016-1662-0](https://doi.org/10.1007/s11390-016-1662-0).
- [3] P. F. Patel-Schneider, "A framework for ontology extraction from databases," in *Proceedings of the International Workshop on Ontology Learning*, Springer, 2005. DOI: [10.1007/11516172\\_12](https://doi.org/10.1007/11516172_12).
- [4] R. Meersman, A. L. de Moor, and H. W. de Bruijn, "Ontology-based xml data management," *Data Knowledge Engineering*, vol. 55, no. 1, pp. 1–10, 2005. DOI: [10.1016/j.datak.2004.11.005](https://doi.org/10.1016/j.datak.2004.11.005).
- [5] S. G. J. Zeng and H. M. Xie, "Ontology extraction from vocabularies and knowledge bases: A survey and new method," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2267–2280, 2015. DOI: [10.1109/TKDE.2014.2345382](https://doi.org/10.1109/TKDE.2014.2345382).
- [6] A. Jiomekong, A. Oelen, S. Auer, L. Anna-Lena, and V. Lars, "Food information engineering," *AI Magazine*, 2023. DOI: [10.1002/aaai.12185](https://doi.org/10.1002/aaai.12185).
- [7] H. T. Azanzi Jiomekong Martins Folefac, "Food composition knowledge extraction from scientific literature," in *Artificial Intelligence: Towards Sustainable Intelligence, AI4S 2023*, S. Tiwari, F. Ortiz-Rodríguez, S. Mishra, E. Vakaj, and K. Kotecha, Eds., ser. Communications in Computer and Information Science, vol. 1907, Springer, Cham, 2023, pp. 89–103, ISBN: 978-3-031-47996-0. DOI: [10.1007/978-3-031-47997-7\\_7](https://doi.org/10.1007/978-3-031-47997-7_7). [Online]. Available: [https://doi.org/10.1007/978-3-031-47997-7\\_7](https://doi.org/10.1007/978-3-031-47997-7_7).
- [8] G. C. Azanzi Jiomekong Hippolyte Tapamo, "An ontology for tuberculosis surveillance system," in *Iberoamerican Knowledge Graphs and Semantic Web Conference*, Springer Nature Switzerland, 2023, pp. 1–15.
- [9] H. B. Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," in *The Semantic Web – ISWC 2023: 22nd International Semantic Web Conference, Athens, Greece, November 6–10, 2023, Proceedings, Part I*, Springer, 2023, pp. 408–427. DOI: [10.1007/978-3-031-47240-4\\_22](https://doi.org/10.1007/978-3-031-47240-4_22).
- [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] D. Jacob, C. Ming-Wei, L. Kenton, and T. Kristina, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2019, pp. 4171–4186.

# DaSeLab at LLMs4OL 2024 Task A: Towards Term Typing in Ontology Learning

Adrita Barua<sup>\*</sup> , Sanaz Saki Norouzi<sup>\*</sup> , and Pascal Hitzler 

Kansas State University, USA

<sup>\*</sup>Correspondence: Adrita Barua, [adrita@ksu.edu](mailto:adrita@ksu.edu)

**Abstract:** The report presents the evaluation results of our approach in the LLM4OL Challenge, where we fine-tuned GPT-3.5 for Task A (Term Typing) across three different datasets. Our approach demonstrated consistent and robust performance during few-shot testing, achieving top rankings in several datasets and sub-datasets, proving the potential of fine-tuning LLMs for ontology creation tasks.

**Keywords:** LLM, Term Typing, Ontology Learning

## 1 Introduction

Large language models (LLMs) have made notable advancements in various natural language processing (NLP) tasks. In a recent study [1], the performance of LLMs was evaluated specifically for Ontology Learning (OL) using zero-shot prompting method. OL refers to the process of creating an ontology—a structured representation of knowledge in a particular domain, consisting of concepts, relationships, and categories. Researchers tackling the challenge of creating ontologies from text are essentially leveraging a broad range of methodologies developed in computational linguistics. By carefully selecting different NLP techniques, they address the three key issues in ontology construction: term association, the creation of term and concept hierarchies, and the identification and labeling of ontological relationships [2].

In [1], they provided an evaluation of three key OL tasks, one of which is term typing. Term typing aims to identify relevant terms from the text that will form the basic vocabulary of the ontology. This task is crucial because it determines the basic building blocks that will be used to construct the ontology. In many respects, ontology learning is a specialized extension of fundamental computational linguistics goals like automatic lexicon construction and semantic text labeling.

As part of the LLM4OL challenge<sup>1</sup> at the International Semantic Web Conference (ISWC) 2024 [3], we focused on fine-tuning GPT-3.5 for term typing across different datasets. The goal was to evaluate the performance of these models during the few-shot testing phase, where the testing dataset includes data from the same ontology domain that the model was trained on. This approach aims to enhance the model's

<sup>\*</sup>These authors contributed equally to this work.

<sup>1</sup><https://sites.google.com/view/llms4ol/home?authuser=0>

ability to accurately identify and categorize relevant terms, thereby improving the overall quality and utility of the created ontology. In the following, first, we talk about the approach and the datasets we use for this task, and then we go through the results and challenge leaderboard, and the conclusion.

## 2 Approach

Our approach involved using three different datasets to individually fine-tune the gpt-3.5-turbo-0125 model<sup>2</sup>, training it to identify term types specific to each dataset. The three fine-tuned models were then evaluated during the few-shot testing phase using their respective test datasets.

### 2.1 Datasets

As part of the LLM4OL challenge [4], we used three datasets to fine-tune the GPT model: WordNet, GeoNames, and UMLS.

**WordNet:** The WordNet dataset is a lexicosemantic dataset derived from the original WordNet. The training set contains 40,559 terms, and the test set has 9,470 terms, covering 18 relation types and four term types: nouns, verbs, adverbs, and adjectives [5].

**GeoNames:** GeoNames consists of geographical locations that comprise 680 categories of geographical locations (e.g., streams, lakes, seas, roads, railroads, etc.). The training set contains 8,078,865 terms, and the test set has 702,510 terms. We used only the first 10 percent (approximately 878,137 terms) of the training dataset to fine-tune our model due to OpenAI's fine-tuning restrictions on the size of the training dataset [6].

**UMLS:** The UMLS (Unified Medical Language System) dataset integrates various biomedical terminologies and standards to support interoperability between different health information systems [7]. Three subcategories of this dataset have been used:

*NCI:* NCI is a UMLS subontology from NCI Enterprise Vocabulary Services (EVS), standardizing terminology for clinical care, research, and public information. It provides reference terminology for NCI and other systems. The training set contains 96,177 terms, and the test set has 24,045 terms, containing 125 term types.

*MEDCIN:* MEDCIN is a UMLS subontology that includes medical components like symptoms and treatments. It uses clinical hierarchies to link data elements, emphasizing relationships within diagnostic contexts. The training set contains 277,028 terms, and the test set has 69,258 terms, containing 87 term types.

*SNOMEDCT\_US:* SNOMEDCT\_US is a UMLS subontology foundational for electronic health records (EHRs), providing concepts with distinct meanings and formal definitions structured hierarchically. The training set contains 278,374 terms, and the test set has 69,594 terms, containing 125 term types.

A detailed discussion of the datasets can be found on the challenge website<sup>3</sup> and the dataset statistics are presented in Table 1.

---

<sup>2</sup><https://platform.openai.com/docs/models/gpt-3-5-turbo>

<sup>3</sup><https://sites.google.com/view/llms4ol/task-a-term-typing?authuser=0>



**Table 1. Dataset Statistics for Fine-Tuning**

Dataset	Training Set	Test Set	No of Term Types	Term Types
<b>WordNet</b>	40,559 terms	9,470 terms	4	Nouns, verbs, adverbs, adjectives
<b>GeoNames</b>	878,137 terms	702,510 terms	680	Geographical locations
<b>NCI</b>	96,177 terms	24,045 terms	125	Clinical care, research, public information
<b>MEDCIN</b>	277,028 terms	69,258 terms	87	Symptoms, treatments
<b>SNOMEDCT US</b>	278,374 terms	69,594 terms	125	Electronic health records (EHRs)

## 2.2 Model Fine-Tuning

In our work, we fine-tuned the gpt-3.5-turbo-0125 model for each dataset using the OpenAI API. Fine-tuning OpenAI’s text generation models is a powerful method to tailor them to specific needs, but it requires significant time and resources [8]. In previous work [1], the authors used a zero-shot prompting method for the term typing task on the aforementioned datasets. We built upon their work by incorporating the optimized prompts from their study into our fine-tuning process to achieve the best results.

To fine-tune the model, we had to prepare the training dataset. For the OpenAI API, the data must be stored in JSONL format, which is a text format where each line is a separate JSON object. This format is ideal for processing large datasets line by line. To prepare the dataset, we created a diverse set of demonstration conversations that closely resemble the interactions the model will encounter during inference in production. Each example in the dataset is formatted as a conversation in the same style as required by the Chat Completions API. Specifically, each example is a list of messages where each message has a role and content. The prompt template can be found in Table 2.

**Table 2. Prompt template for each dataset**

Dataset	Prompt Template
<b>WordNet</b>	Perform a sentence completion on the following sentence: The part of speech of the term "Term" in the sentence "Sentence" is —
<b>GeoNames</b>	Perform a sentence completion on the following sentence: "Place Name/Location" geographically is a ___
<b>UMLS</b>	Perform a sentence completion on the following sentence: "Medical related term" in medicine can be described as ___

The example formats used for generating all three datasets for fine-tuning are given in Tables 3 to 6.

Each entry in the training datasets is formatted according to the example format, using the respective prompt for that dataset to prepare the final JSONL file. After creating all three training datasets, we uploaded the training files to fine-tune the gpt-3.5-turbo-0125 model. Later, the fine-tuned models for each dataset were evaluated using the test datasets. Table 7 highlights the training details of five datasets used for fine-tuning. GeoNames has the largest number of trained tokens (37.7 million) but also the highest training loss (0.0603), indicating that it was more challenging for the model

**Table 3.** Data Format with an Example Sentence (WordNet)

Role	Content
user	Perform a sentence completion on the following sentence: The part of speech of the term "cover" in the sentence "cover her face with a handkerchief" is ___
assistant	The part of speech of the term "cover" in the sentence "cover her face with a handkerchief" is verb.

**Table 4.** Data Format without an Example Sentence (WordNet)

Role	Content
user	Perform a sentence completion on the following sentence: The part of speech of the term "land reform" is ___
assistant	The part of speech of the term "land reform" is noun.

**Table 5.** GeoNames Example

Role	Content
user	Perform a sentence completion on the following sentence: "Pic de Font Blanca" geographically is a ___
assistant	"Pic de Font Blanca" geographically is a peak.

**Table 6.** UMLS Example

Role	Content
user	Perform a sentence completion on the following sentence: "1,2-Dihydro-3-methyl-benz(j)aceanthrylene" in medicine can be described as ___
assistant	The type of "1,2-Dihydro-3-methyl-benz(j)aceanthrylene" in medicine can be described as: ['organic chemical', 'hazardous or poisonous substance'].

to learn from this dataset. WordNet, with the smallest dataset (2.2 million tokens) and a smaller batch size, shows a relatively high training loss (0.0413). In contrast, MEDCIN and SNOMEDCT exhibit the lowest training losses (0.0055 and 0.0086, respectively), suggesting better model performance during training. All datasets were trained for just one epoch, and the varying batch sizes (from 27 for WordNet to 128 for GeoNames, SNOMEDCT, and MEDCIN) reflect the differences in dataset sizes and computational strategies used. Overall, the datasets with larger token counts and higher batch sizes performed well, but some (like GeoNames) may require further tuning to improve performance.

**Table 7.** Training Information for Datasets

Dataset	Trained Tokens	Epochs	Batch Size	LR Multiplier	Training Loss
WordNet	2,208,173	1	27	2	0.0413
GeoNames	37,737,184	1	128	2	0.0603
NCI	6,109,613	1	64	2	0.0273
SNOMEDCT	18,533,107	1	128	2	0.0086
MEDCIN	19,256,674	1	128	2	0.0055

### 3 Evaluation Results

The performance of our fine-tuned models was evaluated across five different datasets. We used the OpenAI API for evaluation, employing the same prompts that were used during the training phase (e.g., as mentioned in the example format for the user’s role 2.2). Each of the three fine-tuned models was assessed using the few-shot testing dataset specific to that model. The results, as provided by the challenge organizers, are summarized in the following tables, which show the leaderboard rankings and the corresponding performance metrics for each dataset. The source code for training and evaluating the models is available online.<sup>4</sup>

#### 3.1 WordNet

Table 8 shows the leaderboard rankings and performance metrics for the WordNet dataset, Our model achieved a top-3 ranking, demonstrating competitive performance in terms of accuracy and other relevant metrics. Here, our model’s performance highlights its effectiveness in achieving balanced precision and recall.

**Table 8.** SubTask A.1 (FS) – Term Typing – WordNet

	Team Name	F1	P	R
1	TSOTSALearning	0.9938	0.9938	0.9938
2	DSTI	0.9716	0.9716	0.9716
3	<b>DaSeLab</b>	<b>0.9697</b>	<b>0.9689</b>	<b>0.9704</b>
4	RWTH-DBIS	0.9446	0.9446	0.9446
5	TheGhost	0.9392	0.9389	0.9395
6	Silp_nlp	0.9037	0.9037	0.9037
7	Phoenixes	0.8158	0.7689	0.8687

#### 3.2 GeoNames

Table 9 presents the leaderboard for GeoNames. Our model secured the first position indicating its superior performance. It’s important to note that our model was evaluated on a portion of the test data, which highlights its robustness and effectiveness even with partial data.

**Table 9.** SubTask A.2 (FS) – Term Typing – GeoNames

	Team Name	F1	P	R
1	<b>DaSeLab</b>	<b>0.5906</b>	<b>0.5906</b>	<b>0.5906</b>
2	Silp_nlp	0.4433	0.7503	0.3146
3	RWTH-DBIS	0.4355	0.4355	0.4355
4	TSOTSALearning	0.2937	0.2937	0.2937
5	TheGhost	0.1489	0.1461	0.1519

#### 3.3 UMLS

As mentioned, this dataset consists of three sub-datasets, and our model demonstrated outstanding performance, ranking first in two of the sub-datasets and second in the other one. The detailed results are presented in the following.

<sup>4</sup><https://github.com/AdritaBarua/LLMs4OL-2024-Task-A-Term-Typing>

### 3.3.1 NCI

In this sub-dataset our model achieved the top ranking, significantly outperforming other models in terms of precision, recall, and F1-score that are shown in Table 10.

**Table 10.** SubTask A.3 (FS) – Term Typing – NCI subontological source from UMLS

	Team Name	F1	P	R
1	<b>DaSeLab</b>	<b>0.8249</b>	<b>0.8161</b>	<b>0.8340</b>
2	Silp_nlp	0.6974	0.8792	0.5779
3	TheGhost	0.5370	0.4450	0.6769
4	RWTH-DBIS	0.1691	0.1821	0.1579
5	Phoenixes	0.0737	0.0562	0.1070

### 3.3.2 SNOMEDCT\_US

Our model also ranked first here, demonstrating its robustness and consistent high performance. The leaderboard is shown in Table 11.

**Table 11.** SubTask A.3 (FS) – Term Typing – SNOMEDCT\_US subontological source from UMLS

	Team Name	F1	P	R
1	<b>DaSeLab</b>	<b>0.8829</b>	<b>0.8810</b>	<b>0.8848</b>
2	Silp_nlp	0.7552	0.8583	0.6742
3	TheGhost	0.5275	0.4266	0.6910
4	RWTH-DBIS	0.4747	0.4888	0.4613

### 3.3.3 MEDCIN

As shown in Table 12, in this sub-dataset, we were ranked as the second one, closely following the top-ranked model. These results indicate that our model maintains a strong balance between precision and recall.

**Table 12.** SubTask A.3 (FS) – Term Typing – MEDCIN subontological source from UMLS

	Team Name	F1	P	R
1	Silp_nlp	0.9382	0.9591	0.9181
2	<b>DaSeLab</b>	<b>0.9373</b>	<b>0.9379</b>	<b>0.9366</b>
3	TheGhost	0.5328	0.4183	0.7336
4	RWTH-DBIS	0.4566	0.4607	0.4526

Analysis of the evaluation shows that the model exhibits significant performance variation across different datasets, particularly with GeoNames demonstrating substantially lower scores compared to WordNet and UMLS datasets. The model achieved an F1 score of 0.5906 on GeoNames, which may be due to the complexity and ambiguity associated with geographical locations and the high number of term types (680), showing a more significant challenge in classification. This ambiguity may refer to the same geographical term representing different places, such as cities with identical names in different countries, or it may arise from varying interpretations of boundaries and regions across cultures and languages. In contrast, WordNet, with its limited scope of four grammatical term types, allowed the model to perform much better, with an F1 score of 0.9697. UMLS datasets, with term types ranging from 87 to 125, still show relatively high scores due to medical terminology’s structured and specialized nature.

## 4 Conclusion

In this paper, we presented the results of our approach to the challenge on different datasets: WordNet, GeoNames, and UMLS. Our models consistently demonstrated robust and competitive performance, achieving top rankings in several datasets and sub-datasets highlighting their strength and potential for practical applications. We are optimistic about the future development and improvement of our approach by utilizing different prompting methods and LLMs.

## Author Contributions

**Adrita Barua:** Coding, Analysis, Writing.

**Sanaz Saki Norouzi:** Coding, Analysis, Writing.

**Pascal Hitzler:** Writing - Review & Editing, Supervision.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements




The authors acknowledge support by the National Science Foundation under awards 2333532 *Proto-OKN Theme 3: An Education Gateway for the Proto-OKN* and 2333782 *Proto-OKN Theme 1: Safe Agricultural Products and Water Graph (SAWGraph): An OKN to Monitor and Trace PFAS and Other Contaminants in the Nation's Food and Water Systems*.

## References

- [1] H. B. Giglou, J. D'Souza, and S. Auer, "LLMs4OL: Large language models for ontology learning," in *The Semantic Web – ISWC 2023 – 22nd International Semantic Web Conference, Athens, Greece, November 6-10, 2023, Proceedings, Part I*, T. R. Payne, V. Presutti, G. Qi, et al., Eds., ser. Lecture Notes in Computer Science, vol. 14265, Springer, 2023, pp. 408–427.
- [2] P. Buitelaar, P. Cimiano, and B. Magnini, *Ontology Learning from Text: Methods, Evaluation and Applications* (Frontiers in Artificial Intelligence and Applications). IOS Press, Amsterdam, 2005, vol. 123.
- [3] 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.
- [4] 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.
- [5] G. A. Miller, "WordNet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [6] GeoNames, *Geonames geographical database*, <http://www.geonames.org/>, 2024.
- [7] O. Bodenreider, "The Unified Medical Language System (UMLS): Integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl\_1, pp. D267–D270, 2004.

- [8] J. Wei, M. Bosma, V. Y. Zhao, *et al.*, "Finetuned language models are zero-shot learners," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022. [Online]. Available: <https://openreview.net/forum?id=gEZrGCozdqR>.

# The Ghost at LLMs4OL 2024 Task A: Prompt-Tuning-Based Large Language Models for Term Typing

Thiti Phuttaamart<sup>1</sup>, Natthawut Kertkeidkachorn<sup>2</sup>, and  
Areerat Trongratsameethong<sup>1</sup>

<sup>1</sup>Chiang Mai University, Chiang Mai, Thailand

<sup>2</sup>Japan Advanced Institute of Science and Technology, Japan

\*Correspondence: Thiti Phuttaamart, [thiti\\_ph@cmu.ac.th](mailto:thiti_ph@cmu.ac.th)

**Abstract:** The LLMs4OL Challenge @ ISWC 2024 aims to explore the intersection of Large Language Models (LLMs) and Ontology Learning (OL) through three main tasks: 1) *Term Typing*, 2) *Taxonomy Discovery* and 3) *Non-Taxonomic Relation Extraction*. In this paper, we present our system’s design for the term typing task. Our approach utilizes automatic prompt generation using soft prompts to enhance term typing accuracy and efficiency. We conducted experiments on several datasets, including WordNet, UMLS, GeoNames, NCI, MEDCIN, and SNOMEDCT\_US. Our approach outperformed the baselines on most datasets, except for GeoNames, where it faced challenges due to the complexity and specificity of this domain, resulting in substantially lower scores. Additionally, we report the overall results of our approach in this challenge, which highlight its promise while also indicating areas for further improvement.

**Keywords:** Large Language Models, Ontology Learning, Prompt Tuning

## 1 Introduction

Currently, most information on the World Wide Web is in a format that is readable and understandable by humans, but computers require significant processing to comprehend this data. To address this, the Semantic Web has been introduced, extending the capabilities of the World Wide Web to make information on the internet interpretable and interconnected more efficiently. This is achieved using Ontology, which models the concepts of information within a specific domain. Typically, creating an ontology is complex, time-consuming, and requires domain expertise. Therefore, Ontology Learning, which automates the extraction and creation of structured data from unstructured information, has been employed. Given the rapid development of Large Language Models (LLMs) with their deep understanding of language, the LLMs4OL Challenge [1] aims to explore and utilize these models to facilitate automatic ontology creation. The LLMs4OL Challenge comprises three tasks.

1. Term Typing: Discover the generalized type for a lexical term
2. Taxonomy Discovery: Discover the taxonomic hierarchy between type pairs

### 3. Non-Taxonomic Relation Extraction: Identify non-taxonomic and semantic relations between types

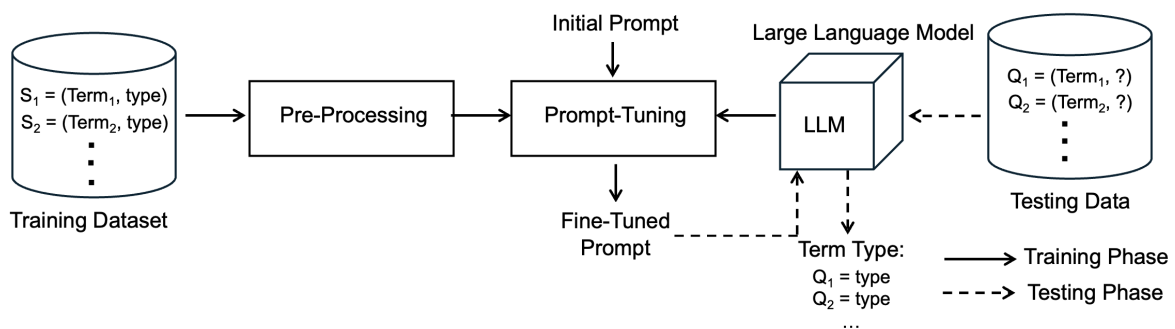
In this study, we are participating in term typing task. The goal of Term Typing task is to assign types to lexical terms. For instance, given the term TUXIS POND from the GeoNames dataset, the correct type would be "lake". For the term typing task, previous methods have primarily focused on using prompts with specific templates to identify term types. However, the key challenge lies in finding an effective prompt that produces accurate results. To address this issue, we propose a prompt-tuning-based LLM for term typing, utilizing automatic prompt generation with soft prompts to enhance both the accuracy and efficiency of the task. The repository of our approach is publicly available (<https://github.com/themes12/Prompt-Tuning-for-LLMs4OL>).

## 2 Related Work

Ontology learning is a technique used to extract knowledge from unstructured text and create structured data known as an ontology. Popular ontology learning methods include using lexico-syntactic patterns [2] and clustering methods [3], or employing lexico-syntactic patterns for term and relation extraction and clustering methods for type discovery [4]. Additionally, seed-term-based bootstrapping methods are also employed [5]. Recently, LLMs have been utilized in ontology learning and have produced promising results [6]. Nevertheless, this method relies on using specific hard prompts, which are difficult to craft and may not yield optimal results. To address these challenges, soft prompting techniques, such as prompt tuning [7], have been developed. Soft prompts involve creating learnable vectors, often referred to as virtual tokens, that are prepended to the input embeddings and further refined through training. Unlike hard prompts, soft prompts do not require manual crafting, making them more flexible and easier to adapt to different tasks.

## 3 Approach

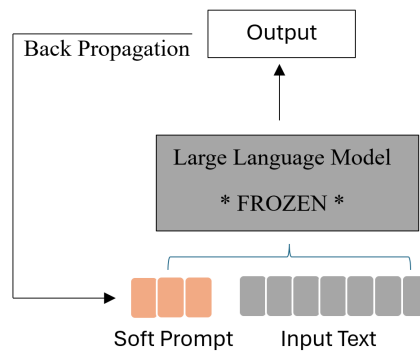
We designed the system, which consists of two phases as shown in Figure 1: 1) Training and 2) Testing. In the training phase, we begin with a dataset containing terms and



**Figure 1.** The design of the system

their types. The data are preprocessed to remove any characters that might cause issues, and then combined with an initial prompt. This input is fed into the LLM to create a fine-tuned prompt. During the testing phase, the fine-tuned prompt is used on new, unseen data, where the terms have no specified types. The LLM predicts the most appropriate type for each term, and the results are formatted for evaluation, ensuring





**Figure 2.** Prompt tuning.

accuracy and alignment with the expected output format. The details of each phase are as follows:

### 3.1 Training Phase

The training phase involves two main steps. The details of each step are as follows:

#### 3.1.1 Pre-processing

The objective of this preprocessing step is to remove characters that could interfere with the final output, particularly during the process of splitting the output by commas to convert it into a list for multi-label classification. This step is essential for ensuring compatibility with the AutoModelForCausalLM and the evaluation system. For instance, in datasets like Gene Ontology, some labels contain commas (e.g., "regulation of alternative mRNA splicing, via spliceosome"), which could disrupt the output if not handled correctly. By removing problematic characters, we can prevent such issues and maintain the integrity of the results. Additionally, the inputs are restructured into a format suitable for training by tokenizing and padding to ensure uniform input length.

#### 3.1.2 Soft Prompting

The objective of the soft prompting step is to efficiently adapt LLMs to perform specific downstream tasks without the need to retrain the entire model for each task. Training LLMs requires a significant amount of time and resources. One effective way to enable a LLM to perform specific downstream tasks is through the use of prompts. Prompts help to describe the task or provide examples of the task (few-shot). There are two types of prompts.

1. Hard prompt involves manually creating the prompt by hand. The downside is that it requires substantial effort to create a good prompt.
2. Soft prompt involves creating a vector, referred to as virtual tokens, and prepend them to the input embeddings for further training with the dataset. The drawback is that humans cannot read the prompt.

In this study, we employ soft prompt. There are various techniques for creating soft prompts, each designed for different tasks. For example, prefix tuning was designed for natural language generation tasks, while P-tuning is designed for natural language understanding tasks. Multitask prompt tuning is another technique that learns a single prompt from data for multiple task types. We have chosen to use the prompt tuning technique because it was initially developed for text classification tasks on T5 models.

This makes it particularly well-suited for our application, as it leverages the strengths of prompt-based methods in handling classification tasks efficiently. The process begins with an initial prompt, which provides a basic template or instruction set for the task. This initial prompt is then refined and adapted during the training process to become a fine-tuned prompt. The advantage of using prompts is that there is no need to train a separate model for each downstream task. Instead, a single LLM can be utilized, greatly reducing the required time.

### 3.2 Testing Phase

The testing phase is designed to evaluate the performance of the fine-tuned LLM on new, unseen data. This phase involves feeding the model with testing data and analyzing its output to determine its accuracy and effectiveness in predicting term types. Once the testing data is prepared, it is fed into the fine-tuned prompt and subsequently into the LLM. The model processes the input terms and generates predictions for their types. The fine-tuned prompt guides the model to understand the context and requirements of the task, leveraging the knowledge gained during the training phase.

## 4 Experiment

### 4.1 Datasets

The datasets used in the term typing task [8] consist of the following four sub-datasets:

1. **WordNet.** WordNet is a lexicosemantics dataset derived from the original WordNet. It contains 40,943 terms for training and 9,470 terms for testing, encompassing four types: nouns, verbs, adverbs, and adjectives.
2. **GeoNames.** GeoNames includes data on geographical locations, with 8,078,865 instances for training, 702,510 instances for testing, and a total of 680 classes.
3. **UMLS.** The UMLS dataset comprises three sub-datasets:
  - **NCI.** Created by NCI Enterprise Vocabulary Services (EVS) to standardize vocabulary for organizational and public use. It includes terms related to clinical care, translational and basic research, public information, and administrative activities, with 96,177 instances for training and 24,045 for testing, covering 125 classes.
  - **MEDCIN.** Contains medical terminology such as symptoms, medical history, physical examination findings, diagnostic tests, diagnoses, and treatment options, with 277,028 instances for training and 69,258 for testing, spanning 87 classes.
  - **SNOMEDCT\_US.** A foundational general terminology used in electronic health records (EHRs), with 277,028 instances for training and 69,258 for testing, encompassing 87 classes.
4. **Gene Ontology.** This dataset includes three sub-ontologies:
  - **Biological Process.** Describes biological processes occurring in living organisms at the cellular level, with 195,775 instances for training and 108,300 for testing, across 792 classes.
  - **Cellular Component.** Describes the positions or structures within a cell, with 228,460 instances for training and 126,485 for testing, covering 323 classes.

**Table 1.** MAP@1 Scores for Our Approach Compared to the Baseline Across Datasets

	WordNet	GeoNames	NCI	MEDCIN	SNOMEDCT_US
Baseline	0.9170	<b>0.4330</b>	0.3280	0.5180	0.4340
Our Approach	<b>0.9368</b>	0.3863	<b>0.6009</b>	<b>0.7397</b>	<b>0.6707</b>

- **Molecular Function.** Describes the activities of gene products, with 196,074 instances for training and 107,432 for testing, spanning 401 classes.

After that, we split the data into 90% for training and 10% for validation in the WordNet, UMLS, and Gene Ontology datasets. For the GeoNames dataset, due to its large size, we split the data into 99% for training and 1% for validation. During the prompt tuning process, the UMLS and Gene Ontology datasets are sampled to 50,000 instances, and the GeoNames dataset is sampled to 100 instances per class. The entire WordNet dataset is used as it is.

## 4.2 Experimental Setup

Our study investigates a range of LLMs, including BLOOM-1B7, BLOOM-3B, BLOOM-7B1, LLaMA-7B, LLaMA-2-7B-HF, LLaMA-2-7B-CHAT-HF, Meta-Llama-3-8B, Meta-Llama-3-8B-Instruct, BioMistral-7B, and LLaMA-OpenBioLLM-8B. Based on the results from the validation datasets, we selected the following models for each dataset: BLOOM-3B for WordNet, NCI, and SNOMEDCT\_US; Meta-Llama-3-8B-Instruct for GeoNames and Biological Process; BLOOM-1B7 for MEDCIN; and BioMistral-7B for Cellular Component and Molecular Function. We implemented the models using AutoModelForCasualLM and set the hyperparameters as follows: learning rate:  $3e - 2$ , epochs: 2-4, train size: 15% for WordNet and 5% for GeoNames, and 30% for other datasets. The max token length is 10, and the virtual token size is 15 for WordNet, 40 for GeoNames, 30 for UMLS, 30 for Biological Process and Molecular Function, and 29 for Cellular Component. The choice of models and hyperparameters is based on the results obtained from experiments on the validation datasets <sup>1</sup>.

We used the best results presented in the study [6] as the baseline. Please note that only WordNet, UMLS, GeoNames, NCI, MEDCIN, and SNOMEDC\_US were investigated. For evaluation metrics, we use MAP@1 (Mean Average Precision at rank 1) [6] to compare our results with the baseline. MAP@1 measures the precision of the top-ranked result for each query, providing an assessment of the model's effectiveness in retrieving the most relevant results. For reporting the results of our approach on this challenge, we use the standard metrics of precision, recall, and F1 score as provided by the challenge organizers.

## 5 Result and Discussion

Table 1 presents the MAP@1 scores for our approach compared to the baseline, using the same datasets and evaluation metrics as described in LLMs4OL: Large Language Models for Ontology Learning [6]. Our approach shows enhanced performance across datasets such as WordNet, NCI, MEDCIN, and SNOMEDCT\_US, indicating improved term retrieval precision. However, the results for GeoNames reveal persistent challenges related to place name ambiguity. The results of the term typing task across different datasets are summarized in Table 2. The results indicate that the system performs well on the WordNet, NCI, SNOMEDCT\_US, and MEDCIN datasets. However, in

<sup>1</sup> <https://github.com/themes12/Prompt-Tuning-for-LLMs4OL/blob/main/result-validation.pdf>

**Table 2.** The result on term typing task

Dataset	F1	Precision	Recall
WordNet	0.9392	0.9389	0.9395
GeoNames	0.1489	0.1461	0.1519
NCI	0.5370	0.4450	0.6769
MEDCIN	0.5328	0.4183	0.7336
SNOMEDCT_US	0.5275	0.4266	0.6910
Cellular Component	0.1877	0.1653	0.2171
Biological Process	0.1025	0.0964	0.1095
Molecular Function	0.1270	0.1278	0.1261

the NCI, SNOMEDCT\_US, and MEDCIN datasets, the recall is significantly higher than the precision, which may be due to class imbalance. The performance on the GeoNames and Gene Ontology datasets is significantly worse. For GeoNames, the problem may stem from the ambiguity of place names and the fact that these names are often proper nouns, making them difficult to predict. Additionally, datasets like the Biological Process dataset, which has 792 classes, or the Geonames dataset, with 680 classes, are more challenging compared to smaller datasets like WordNet, which has only 4 classes, or the NCI dataset, with 125 classes. The larger number of classes in these bigger datasets can make predictions harder. For the Gene Ontology dataset, the poor results may be due to the biological nature of the data, which includes information on genes, molecules, and structures. This domain is highly specialized and contains a vast number of possible classes.

## 6 Conclusion

In this study, we explored the use of soft prompt tuning for the term typing task as part of the LLMs4OL Challenge @ ISWC 2024. Our approach demonstrated strong performance on several datasets, particularly WordNet and UMLS sub-datasets (NCI, MEDCIN, SNOMEDCT\_US), indicating the viability of soft prompt tuning for ontology learning tasks. However, the results on GeoNames and Gene Ontology datasets were less satisfactory, highlighting challenges such as class imbalance and the complexity of specialized domains. To improve the results, future work could focus on incorporating additional contextual information beyond just the term, which may help the LLM make better predictions. Additionally, employing techniques other than soft prompts, such as Retrieval-Augmented Generation (RAG), could enhance the LLM's ability to access up-to-date knowledge and external information, potentially leading to improved prediction capabilities. These strategies could address the current limitations and further advance the effectiveness of soft prompt tuning for ontology learning tasks.

## Author contributions

**Thiti Phuttaamart:** Software; Writing – Original Draft Preparation; Conceptualization.  
**Natthawut Kertkeidkachorn:** Conceptualization; Writing - Review Editing; Project administration; Supervision.  
**Areerat Trongratsameethong:** Writing - Review Editing; Project administration; Supervision.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgement

This work was supported by JSPS Grant-in-Aid for Early-Career Scientists (Grant Number 24K20834).

## References

- [1] 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.
- [2] M. Hearst, *Automated discovery of wordnet relations.* wordnet an electronic lexical database, 1998.
- [3] L. Khan and F. Luo, "Ontology construction for information selection," in *14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings.*, IEEE, 2002, pp. 122–127.
- [4] J. Watróbski, "Ontology learning methods from text-an extensive knowledge-based approach," *Procedia Computer Science*, vol. 176, pp. 3356–3368, 2020.
- [5] C. H. Hwang, "Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information.," in *KRDB*, Citeseer, vol. 21, 1999, pp. 14–20.
- [6] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," in *The Semantic Web – ISWC 2023*, T. R. Payne, V. Presutti, G. Qi, *et al.*, Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.
- [7] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.
- [8] 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.

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

## Applications on WordNet and GeoNames Datasets

Hanna Abi Akl<sup>1,2</sup> 

<sup>1</sup>Data ScienceTech Institute (DSTI), 4 Rue de la Collégiale, 75005, Paris, France

<sup>2</sup>Université Côte d'Azur, Inria, CNRS, I3S

\*Correspondence: Hanna Abi Akl, [hanna.abi-akl@dsti.institute](mailto: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<sup>1</sup> 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.

---

<sup>1</sup><https://sites.google.com/view/llms4ol/home>

## 2 Methodology

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

$$ST = \{s_1, s_2, \dots, s_n\}, \quad (1)$$

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<sup>2</sup> 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<sup>3</sup> model by Google [12].
5. The resulting domain vector embeddings are stored in a MongoDB<sup>4</sup> 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

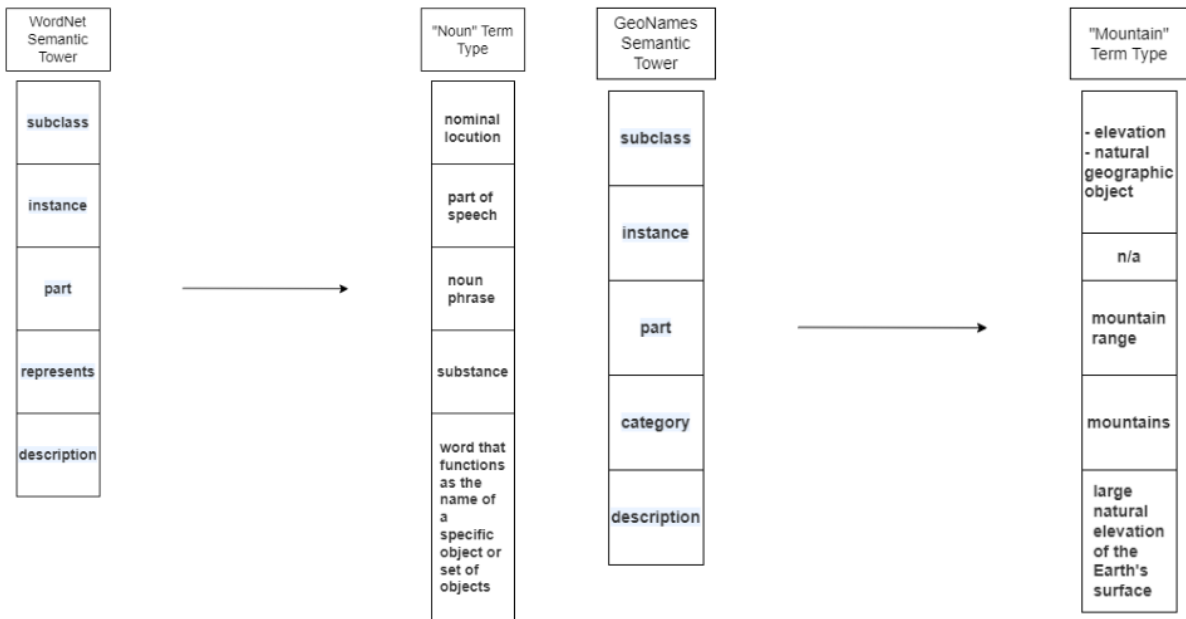
We consider two datasets for our experiments: WordNet and GeoNames. Both datasets are used for training and testing our models in the respective subtasks (A.1 and A.2). The dataset descriptions are detailed in the following subsections.

---

<sup>2</sup><https://query.wikidata.org/>

<sup>3</sup><https://huggingface.co/thenlper/gte-large>

<sup>4</sup><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 Abelles	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`<sup>5</sup> 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 ≥ 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`<sup>6</sup> and `flan-t5-small-geonames`<sup>7</sup>.

### 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.

<sup>5</sup><https://huggingface.co/google/flan-t5-small>

<sup>6</sup><https://huggingface.co/HannaAbiAkl/flan-t5-small-wordnet>

<sup>7</sup><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<sup>8</sup>.

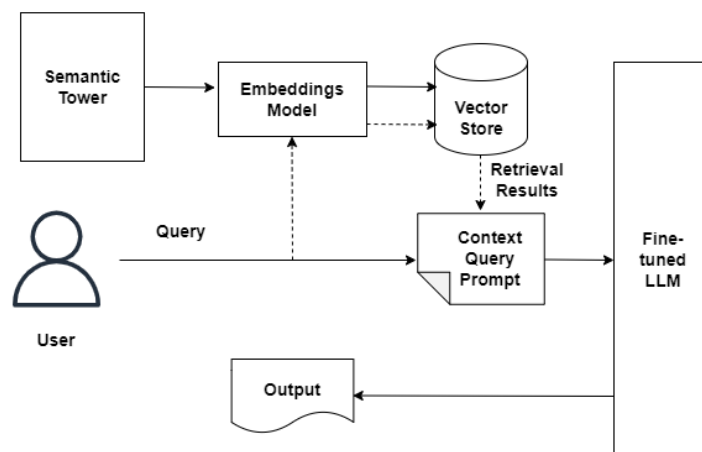


Figure 4. RAG system architecture.

<sup>8</sup><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
<b>flan-t5-small-wordnet (WN1)</b>	<b>0.9820</b>
flan-t5-small-wordnet + WordNet semantic tower (WN2)	0.8581

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

Experiment	F1
<b>flan-t5-small-geonames (GN1)</b>	<b>0.6820</b>
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
TSOTSA Learning	0.9938	0.9938	0.9938
<b>DSTI (WN1)</b>	<b>0.9716</b>	<b>0.9716</b>	<b>0.9716</b>
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
<b>DSTI (WN2)</b>	<b>0.8420</b>	<b>0.8420</b>	<b>0.8420</b>
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. Fährdrich, *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.

# SKH-NLP at LLMs4OL 2024 Task B: Taxonomy Discovery in Ontologies Using BERT and LLaMA 3

Seyed Mohammad Hossein Hashemi<sup>1</sup>, Mostafa Karimi Manesh<sup>1</sup>, and Mehrnoush Shamsfard<sup>1</sup>

<sup>1</sup>Shahid Beheshti University, Tehran, Iran

seyedmo.hashemi@mail.sbu.ac.ir, {m\_karimimanesh,m-shams}@sbu.ac.ir

\*Correspondence: Seyed Mohammad Hossein Hashemi, seyedmo.hashemi@mail.sbu.ac.ir

**Abstract:** Taxonomy discovery in ontologies refers to extracting the parent class from the child class. By modeling this task as a classification problem, we addressed it using two different approaches. The first approach involved fine-tuning the “BERT-Large” model with various prompts and using it in a classification system. In the second approach, we utilized the “LLaMA 3 70B” model, experimenting with different prompts and modifying them to achieve the best results. Additionally, we evaluated the correctness of the answers using substring and Levenshtein distance functions. The results indicate that, with appropriate fine-tuning, the BERT model can achieve performance levels comparable to those of more recent and significantly larger language models, such as LLaMA 3 70B. However, with appropriate prompts, LLaMA 3 70B performs slightly better than BERT, highlighting the importance of prompt quality. Ultimately, further experiments on different settings for fine-tuning BERT, few-shot learning, and using knowledge graphs for validating the model’s answers for LLaMA are recommended to improve the results. Additionally, testing other models and examining the results of various encoder-based and decoder-based models can be employed.

**Keywords:** Large Language Models, LLMs, Ontologies, Ontology Learning, Fine-tuning, Prompting, Prompt-based Learning, BERT, LLaMA 3

## 1 Introduction

One of the applications of large language models (LLMs) is learning ontologies from input text. This process is divided into three sections: *term typing*, *taxonomy discovery*, and *non-taxonomic relationship extraction* [1]. In the LLMs4OL challenge [2], the goal is to develop models to perform these tasks automatically. The task selected by the authors of this paper is Task B, i.e., taxonomy discovery. Among the available datasets in the challenge [3], the GeoNames dataset was chosen for this project. This dataset is extracted from the GeoNames ontology [4], which is a geographical database that provides information about locations and geographical features around the world. This data includes place names, geographic coordinates (latitude and longitude), place types (such as city, village, river, mountain, lake, etc.), elevation above sea level, postal

codes, population, and other attributes. Among the features available in this ontology, the place name along with its type has been extracted.

For solving this problem, the language models BERT-Large and LLaMA 3 70B are being used. The following reasons were involved in choosing these two models:

- BERT performs very well in many traditional NLP tasks such as classification and information extraction.
- In the studies of BabaeiGiglou et al.[1], BERT was able to achieve remarkable results without the need for fine-tuning, with its results in this task being close to other mentioned LLMs like GPT-3.
- Fine-tuning BERT model was doable for the authors of the article given their hardware access.
- LLaMA 3 70B has a larger size and more modern architecture comparing to BERT.

As shown by BabaeiGiglou et al. [1], in Task B with the GeoNames dataset, the best result was achieved by the GPT-3.5 model, with an F1-score of 67.8. Among the fine-tuned models, FLAN-T5 Large obtained the best result with an F1-score of 62.5. The BERT and LLaMA 7B models also reached scores of 54.5 and 33.5, respectively, without fine-tuning. However, these scores were based on the model's performance in binary classification, determining whether there is a relationship between a child and parent or not. In contrast, the task in this project is to identify the parent of each child.

In this project, the competition training dataset was first received, and additional necessary data was generated. Then, with the help of two different approaches using the mentioned language models, various methods were presented for solving the problem. The final results of each method and their analysis are discussed, and finally, ideas for improving these models are provided.

## 2 Augmentation of Training Data

The dataset provided to participants for this challenge includes 476 records of (child, parent) pairs from the GeoNames ontology. In this dataset, the parent column contains 9 distinct values; therefore, this dataset can be considered a classification dataset with 9 classes. To train a classifier using BERT, in addition to this dataset, we also needed a negative dataset, which we generated through the procedure described below.

From the 476 records in the initial dataset, 76 records were separated for validation to ensure no overlap between the training and validation datasets. Then, using a consistent pattern, negative data was generated for both the training and validation datasets. Two approaches were considered for generating negative data:

1. Generating reversed records: Reversed records are records that are exactly copied from the positive dataset, with the parent and child swapped.
2. Generating manipulated records: Manipulated records are also exactly copied from the positive dataset, with one of the other 8 parents randomly replacing the original parent in each record.

The number of records in the generated negative dataset for each dataset equals the number of records in the original dataset. Approximately one-third of the negative dataset consists of reversed records, and two-thirds are manipulated records. For example, for a set of 400 positive records in the training data, 133 reversed records and 267 manipulated records were generated. In the final dataset, we have a set of

records where the positive or negative status is indicated by a column titled "label" that can be True or False.

### 3 Proposed Methods for Taxonomy Discovery

As mentioned earlier, two different approaches were used to solve this problem. One was fine-tuning BERT, and the other was using the LLaMA 3 70B model. Details of each approach are explained below. Before continuing, since the discussed problem can also be considered a classification problem, from here onward, the concepts of class and parent (destination of the is-a relationship) and also instance and child (source of the is-a relationship) are considered synonymous. Additionally, since the main focus is on identifying the class (in classification problem) or the parent (in ontology hierarchy), any mention of class refers to the parent and vice versa.

#### 3.1 BERT-Based Approach

To solve this problem using BERT, we modeled it as a multi-class classification task with 9 classes. The method involves using a single binary classifier iteratively for each of the 9 classes. The classifier determines whether an instance belongs to a specific class or not by receiving the instance and class as inputs. For example, when predicting a child's parent, the classifier first determines if the child belongs to class 1 or not, then class 2, and so on for all subsequent classes.

Other approaches could have been applied, such as using a separate classifier for each class or employing a single 9-class classifier for all classes. Our method in using a single binary classifier is scalable to larger datasets and a greater number of classes without requiring the training of multiple models or designing a separated multi-class classifier when the classes are changed.

We used the BERT-Large model to solve this problem. Initially, this model was fine-tuned on the training dataset as a binary classifier. This means that the final model can determine whether there is an is-a relationship between the given (parent, child) pair or not. To use this model for a 9-class classification problem, we need to check whether a child belongs to a parent once for each parent, and based on the output, the relevant class is extracted. Details of this method are explained below.

During the fine-tuning and testing of the models, an additional prompt (the ninth in the following list) was added to the 8 prompts used by BabaeiGiglou et al. [1], resulting in a total of 9 prompts. The complete set of prompts is as follows:

1. parent is the superclass of child. This statement is [MASK].
2. child is a subclass of parent. This statement is [MASK].
3. parent is the parent class of child. This statement is [MASK].
4. child is a child class of parent. This statement is [MASK].
5. parent is a supertype of child. This statement is [MASK].
6. child is a subtype of parent. This statement is [MASK].
7. parent is an ancestor class of child. This statement is [MASK].
8. child is a descendant class of parent. This statement is [MASK].
9. "parent" is the superclass of "child". This statement is [MASK].

In these prompts, *parent* and *child* are replaced with the appropriate parent and child, and [MASK] is the token that the model needs to predict. In this set, there are 4 superclass statements, 4 corresponding subclass statements, and 1 additional



superclass statement grouped together. This set of prompts has been used in different ways to fine-tune BERT, which will be discussed in section 4.1.

To determine the parent class for a child class, we initially ask the model nine questions, each corresponding to one of the potential parent classes. These questions are posed in the format of the first prompt. If the model answers "True" to only one of these nine questions while answering "False" to the remaining eight, the parent class associated with the "True" response is selected. If no single parent class stands out, we proceed with two additional prompts in sequence. At each stage, only parents with the highest score, meaning those for which the model has returned True for more prompts, advance to the next stage. This process continues until the end of the prompt list. In the final step, if multiple parent classes still have the highest score, the system randomly selects one from these parents. The percentage of instances where random selection was used relative to the total number of instances can be a criterion for evaluating different systems.

### 3.2 LLaMA-Based Approach

After evaluating Task B using the fine-tuned Bert-Large model, it was decided to perform the evaluation using the LLaMA 3 70B as well. In this phase, the focus was mainly on the prompts. The general structure of the prompts follows two main concepts:

1. Classification Concept (instance and class)
2. Hierarchy Concept (is-a) (parent and child)

In the first category, the prompts contain a classification definition, asking the model to identify the class based on the given input. However, in the second category, the problem is defined as an is-a hierarchy, and the model is asked to identify the destination of the relationship (parent) based on the input (child).

After observing the model's responses, one challenge identified was the class names. Each class title is a combination of several terms (e.g., "mountain, hill, rock"). Despite mentioning the class titles in the prompts, in some cases, the model only used part of the class title in its responses. For example, in response to the question about "cattle dipping tank," which corresponds to the class "spot, building, farm," the model only used "spot" as the answer. Given these conditions, during the evaluation phase, in addition to evaluating the model's output separately, the substring function and Levenshtein distance [5] were applied to the model's output. The substring function returns the class title that the output of the model is a part of. The Levenshtein function returns the class title that is closer to the output of the model based on the Levenshtein distance.

In addition to the mentioned actions, to save time, instead of providing samples one by one, a set of samples formatted in a specific way was fed to the model, and responses were received in batches. To manage this issue, each sample was assigned a unique number, and the model was asked to separate the response sections and include the number of each question alongside its response. Subsequent results indicate that batch questions are not as accurate as individual questions.

## 4 Experiments

In this section, we present the experiments conducted on the two groups of systems discussed: BERT-based systems and LLaMA-based systems. Implementations and datasets used in these experiments are available in a GitHub repository<sup>1</sup>.

### 4.1 Experiments on the BERT-Based Systems

In this section, we examine the details of the systems implemented using BERT. All systems are fine-tuned using the methods mentioned in the previous section, with the goal of predicting the parent of each child. Due to time and hardware constraints during the competition, the BERT model was fine-tuned using fixed hyperparameters. Hyperparameter tuning could potentially lead to improved results.

The first category of our systems consists of those where BERT was trained sequentially with 1, 2, 3, ... up to 8 different prompts, with one epoch of training for each prompt. Since the terms *ancestor* and *descendant* used in prompts 7 and 8 differ somewhat from those used in the other prompts, two more systems were trained separately: one on the set of prompts 1 to 4 and 7, and another on prompts 1 to 4, 7, and 8. For the testing phase of all the aforementioned models, 9 prompts were used.

By analyzing the results and performance of the systems, and considering the functioning and structure of BERT, we hypothesized that using one set out of superclass statements and subclass statements could help improve the results. For this purpose, in the next category of systems, BERT was trained only on superclass statements, as follows: once with prompt 1, once with prompts 1 and 3, once with prompts 1, 3, and 5, etc. Each prompt is being trained for one epoch. For the testing phase of these models, the same 5 prompts are used.

Table 1 presents the results of the different systems on the validation dataset. As mentioned, in each system, the predicted parent in some instances was randomly selected from among the candidate parents. The last column of this table indicates the percentage of parents that were *not randomly* selected. The metric values are reported in percentage. These values are rounded to one decimal place in all columns except the last one, where they are reported without decimal places. Additionally, weighted averages were used in calculating precision, recall, and F1-score. In this table, the best result in each column is bold and underlined, and the second and third best results in each column are bold.

### 4.2 Experiments on the LLaMA-Based Systems

The initial results using the prompts on the evaluation dataset, which was submitted for the competition, are presented in Table 2. The values are rounded to one decimal place in all columns.

In both prompts, the classification problem and the is-a relationship were defined precisely:

- "The problem under consideration is classification. X is a subclass of Y, meaning that X shares common features and properties with other members of class Y."
- "If we say "X is a Y," it means that X is a specific instance of Y and inherits all the features and behaviors of Y."

---

<sup>1</sup><https://github.com/s-m-hashemi/llms4ol-2024-challenge>

**Table 1.** Evaluation results of different BERT-based systems on validation dataset.  
\*SC: Superclass Statements

No	Model	Precision	Recall	F1-Score	% of Non-Randoms
1	Prompt 1	6.6	18.4	8.4	71
2	Prompts 1, 2	56.8	22.4	21.9	<b>95</b>
3	Prompts 1-3	44.7	40.8	38.8	<b>92</b>
4	Prompts 1-4	54.2	25.0	23.5	<b>95</b>
5	Prompts 1-5	53.8	44.7	44.3	49
6	Prompts 1-6	45.2	34.2	34.4	66
7	Prompts 1-7	<b>67.5</b>	<b>61.8</b>	<b>63.0</b>	54
8	Prompts 1-8	37.0	25.0	23.1	53
9	Prompts 1-4, 7	63.0	0.5	<b>52.9</b>	62
10	Prompts 1-4, 7, 8	<b>64.7</b>	18.4	12.0	<b>83</b>
11	SC* Prompt 1	8.9	19.7	10.1	71
12	SC Prompts 1, 2	7.2	21.1	10.3	70
13	SC Prompts 1-3	50.5	<b>51.3</b>	45.6	32
14	SC Prompts 1-4	42.2	50.0	45.0	29
15	SC Prompts 1-5	<b>66.2</b>	<b>59.2</b>	<b>60.8</b>	50

**Table 2.** Evaluation results of LLaMA 3 70B tested using different prompts on validation dataset.

Prompt / Eval metrics	Precision			Recall			F1-Score		
	None	Sub	Levn	None	Sub	Levn	None	Sub	Levn
Class concept	64.9	64.8	57.1	51.3	51.3	51.3	54.6	54.6	51.9
Is-a (individual query)	21.4	72.6	47.2	7.8	64.4	21	10.1	62.9	18.7
Is-a (batch query)	0	68.4	16.8	0	39.4	13.1	0	46.4	8.2

An example of the prompts is as follows:

- "If we say 'X is a Y,' it means that X is a specific instance of Y and inherits all the features and behaviors of Y. Given an instance as 'X,' select the most appropriate 'Y' from (city, village) or (country, state, region) or (forest, heath) or (mountain, hill, rock) or (parks, area) or (road, railroad) or (spot, building, farm) or (stream, lake) or (undersea)."

In order to improve the results, several iterations of modifying the prompt definitions were undertaken, leading to improved outcomes, which are detailed below.

In the initial prompts, a sample was included to clarify the definition. For example, in the classification prompt, it was stated: "wadi mouth" is considered a subclass of "parks, area." We observed that the model tended to favor the mentioned class. Based on this observation, this example was removed from the new prompts. Furthermore, the class names, due to their specific structure and the presence of commas between them, needed to be more precisely distinguished. Therefore, each class title was enclosed in a pair of parentheses, and the term "or" was used between them.

In the initial prompt, it was written: "In lexical networks, a concept known as a triplet is discussed. This triplet is formed between two words and a relationship between them." However, in the improved prompt, the definition was changed to: "If we say 'X is a Y,' it means that X is a specific instance of Y and inherits all the features and behaviors of Y." The results with the evaluation data using the modified prompts are presented in Table 3. The values are rounded to one decimal place in all columns.

**Table 3.** Evaluation results of LLaMA 3 70B tested using improved prompts on validation dataset.

Prompt / Eval metrics	Precision			Recall			F1-Score		
	None	Sub	Levn	None	Sub	Levn	None	Sub	Levn
Class concept	75.8	75.7	73.4	64.4	71	67.1	67.7	72	67.7
Is-a	76.2	76	73.8	65.7	72.3	68.4	69.1	73.1	68.9

By examining the results of batch and individual submissions, it was found that the results in the batch mode were weaker, so in this phase, batch question evaluations were not conducted.

### 4.3 Results of the Systems on the Test Dataset

The results of the BERT-based and LLaMA-based systems on the final test dataset are presented in Table 4. This table includes the results of the three best BERT-based and two best LLaMA-based systems, both with the best F1-scores on the validation dataset. In each column, the best result is bold and underlined, and the second-best result is bold. The two best LLaMA-based systems are those mentioned in Table 3 with substring function applied.

However, since the results for the LLaMA-based models in this table are based on a new prompt that was tested after the competition, the best result during the competition was achieved by the BERT-based models.

**Table 4.** Evaluation results of best systems on test dataset.

No	Model	Precision	Recall	F1-Score	% of Non-Randoms
1	Prompts 1-7	67.2	62.7	<b>62.8</b>	<b>56</b>
2	SC Prompts 1-5	<u><b>78.1</b></u>	56.4	62.5	47
3	Prompts 1-4, 7	64.6	47.1	51.4	<u><b>70</b></u>
4	Prompt with class concept	<b>69.4</b>	<b>67.6</b>	<b>66.5</b>	-
5	Prompt with is-a concept	68.0	<b>63.2</b>	62.3	-

## 5 Results Analysis

### 5.1 Analysis of BERT-Based Systems Results

The BERT model, when exposed to various prompts, can learn to focus on the relationship between the two target words rather than other words in the sentence. This ability generally leads to improved results as the number of training prompts increases. However, when subclass statement prompts are introduced, except for the second prompt, performance decreases, as shown in Table 1. Consequently, the improvement trend continues until prompt 7, but with the addition of prompt 8, the results significantly deteriorate. It seems that the sharp decline in results with prompt 8 is due to the different words used in prompts 7 and 8. This pattern is also observed when comparing models 9 and 10, where the inclusion of prompt 8 leads to a noticeable drop in various metrics.

During the system design phase, it was hypothesized that BERT might perform better if it consistently sees the (parent, child) pairs in sentences in a fixed order. Therefore, in systems 12 through 16 in Table 1, only prompts in which the parent comes first and the child second, referred to as superclass statement prompts, were used. Although these systems do not perform well with a small number of prompts, as the number of

prompts increases to three, the results improve significantly, reaching their best with five prompts.

Looking at the last column in Table 1, it is observed that the models with the lowest F1-scores produce the least random results. However, our best models generate 50 to 60 percent of their results randomly. The reason for this is that while the initial models are more confident in their generated answers, the quality of those answers is not sufficient. On the other hand, in many cases, this random selection is made from between two or three parent candidates, which contributes to the better performance of the final systems. Nevertheless, efforts to reduce the percentage of randomly-generated answers could be a focus for future stages.

Examining the results of the systems on the test dataset, as shown in Table 4, also shows that these results are fairly close to the validation dataset results, and the pattern of results across different systems is consistent. This consistency suggests that BERT has been able to generalize significantly even with a relatively small dataset. The best result comes from the system trained with the first seven prompts, achieving an F1-score of 62.8 percent. Close behind is the system trained with five superclass statement prompts, scoring 62.5 percent.

## 5.2 Analysis of LLaMA-Based Results

The following points are noteworthy after reviewing the results obtained from structural changes in the prompts:

- In the initial prompt, where concepts were defined broadly, the class concept performed better than the is-a relationship, which was not well understood by the model.
- Using variable names in the prompt definitions and introducing how each concept fits into the prompt helped in understanding the relationships.
- Since the answers are drawn from a set of nine options, including an example in the prompt tends to bias the model toward that answer.
- Contrary to the initial prompt, where classification results were better than the is-a relationship, after modifications in the prompt definitions, the is-a relationship shows better results.
- After structural changes in the prompts, the results for both classification and is-a methods are very close.
- Applying the substring function on the results derived from the improved prompts increases the F1-score on the evaluation dataset to 73.1, which is notable.
- The results from the improved prompts (which were not used in the competition) on the test data show an F1-score of 66.5.

## 6 Conclusion and Future Work

As observed, BERT, when properly fine-tuned, can yield outstanding results in the taxonomy discovery task in the competition. However, spending more time and experimenting with different combinations of prompts could significantly improve these results. It was also seen that by using appropriate prompts for the LLaMA 3 70B model and adding an auxiliary function like substring, even better results can be achieved. Although this model produces better results than BERT-based systems, the small gap indicates that BERT, when fine-tuned properly, is well-suited for this task. Our results show a significant improvement over the best results reported by BabaeiGiglou et al.

[1] in Task B on the GeoNames dataset, in which the task was simplified as a binary classification problem. This suggests that the methods examined in this paper perform very well in taxonomy discovery.

Future work could explore the following ideas for extending this work.

- For the BERT-based systems:
  - Adding more prompts to the set of prompts.
  - Increasing the number of epochs per prompt while training the model.
  - Using a set of subclass statements instead of superclass statements.
  - Not generating inverted records in the negative dataset.
  - Utilizing other encoder-based language models.
- For the LLaMA-based systems:
  - Using Few-Shot Learning in the prompts and examining its impact on results.
  - Applying the same prompts used in this study to GPT-4 and comparing the results with LLaMA 3 70B.
  - Comparing the results of LLaMA 3 8B and LLaMA 7B.
  - Using knowledge graphs to analyze the relationship between the model's response and the correct answer.

## Authors Contributions

**Seyed Mohammad Hossein Hashemi:** Conceptualization, Methodology, Software, Validation, Investigation, Data Curation, Writing - Original Draft, Project administration.

**Mostafa Karimi Manesh:** Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft.

**Mehrnoush Shamsfard:** Writing - Review & Editing, Supervision.

## Competing Interests

The authors declare that they have no competing interests.

## References

- [1] H. Babaei Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," in *The Semantic Web – ISWC 2023*, T. R. Payne, V. Presutti, G. Qi, *et al.*, Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.
- [2] 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.
- [3] 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.
- [4] "Geonames." (n.d.), [Online]. Available: <https://www.geonames.org/> (visited on 08/05/2024).
- [5] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Proceedings of the Soviet physics doklady*, 1966.