Supplementary Materials for "OntoLearner: A Modular Python Library for Ontology Learning with Large Language Models"

Contents

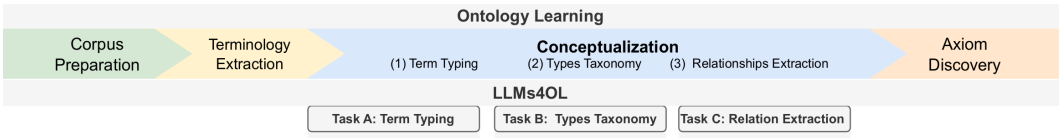


Figure 1: Overview of the OL pipeline and the role of the LLMs4OL paradigm tasks. The workflow spans from corpus preparation and terminology extraction to conceptualization and axiom discovery. Within the conceptualization phase, LLMs4OL focuses on three core tasks: (A) term typing, (B) taxonomy discovery, and (C) non-taxonomic relation extraction. These tasks collectively support the construction and enrichment of ontologies using LLMs.

1 LLMs4OL Paradigm Tasks Definition

In this section, we formalize the core ontology learning (OL) tasks addressed in the LLMs4OL paradigm (see Figure 1) and illustrate each task with concrete examples [6, 18, 19]. These tasks capture fundamental ontology engineering operations and serve as standardized benchmarks for evaluating Large Language Models (LLMs) within OntoLearner, which serves as a real-world use case.

The LLMs4OL paradigm focuses on three primary tasks: (i) term typing, which involves assigning semantic categories or types to domain-specific terms; (ii) taxonomy discovery, which aims to organize these terms into hierarchical structures based on "is-a" relationships; and (iii) non-taxonomic relationship extraction, which identifies meaningful semantic relations beyond hierarchies, such as functional or associative links between concepts. Together, these tasks capture the essential components of ontology learning—concept classification, hierarchical organization, and relational enrichment. They form the foundation of the LLMs4OL evaluation framework, enabling systematic benchmarking of both retrieval-based and generative approaches across diverse domains and datasets.

1.1 Term Typing

Definition. Term typing is defined as the task of assigning one or more semantic types (classes) to a given lexical term. Formally, let $\mathcal{L}$ denote a lexical term and $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$ the set of candidate ontology types. The goal is to predict a subset $\mathcal{T}^*(\mathcal{L}) \subseteq \mathcal{T}$ such that each selected type correctly characterizes the meaning of $\mathcal{L}$.

Example.

- *Input term:* "aspirin"
- *Candidate types:* {Drug, ChemicalCompound, Disease, Procedure}
- *Output:* {Drug, ChemicalCompound}

This is a multi-class classification problem where a term may belong to multiple semantic categories. The task is essential for populating ontologies and enabling downstream reasoning.

1.2 Taxonomy Discovery

Definition. Taxonomy discovery aims to identify hierarchical ("is-a") relationships between concepts. Given two terms or classes (c_i, c_j), the task is to determine whether a subsumption relation holds, i.e., whether $c_i \sqsubseteq c_j$ ("c_i is a subclass of c_j").

Example 1.

- *Input pair:* ("red wine", "wine")
- *Output:* True ("red wine" is a type of "wine")

Example 2.

- *Input pair:* ("wine", "red wine")
- *Output:* False

Table 1: Complexity Score Analysis.

(a) Robustness analysis summary by test type (Spearman ρ vs original ranking)

Test type	**N variants**	**Min ρ**	**Mean ρ**	**Interpretation**
Weight perturbation	20	0.9990	0.9996	Extremely stable to moderate reweighting
Sigmoid parameters	24	1.0000	1.0000	Ranking invariant to tested (a, b) choices
Normalization	2	0.8557	0.8564	Largest sensitivity source
Equal weighting	1	0.9968	0.9968	Family-level priors matter little for ranking

(b) Spearman ρ under family-weight perturbations

Family	**-20%**	**-10%**	**+10%**	**+20%**
Graph	0.9992	0.9997	0.9998	0.9995
Coverage	0.9996	0.9998	0.9998	0.9994
Hierarchy	0.9999	0.9999	1.0000	0.9999
Breadth	0.9990	0.9996	0.9997	0.9991
Dataset	0.9997	0.9998	0.9998	0.9996

This task is typically formulated as a binary classification or ranking problem and is central to constructing class hierarchies. It requires understanding semantic generalization and specialization relationships.

1.3 Non-Taxonomic Relationship Extraction

Definition. This task focuses on identifying semantic relationships between entities that are not hierarchical. Given two entities (e_i, e_j) and a set of possible relations $\mathcal{R}$, the objective is to predict the most appropriate relation $r \in \mathcal{R}$ or determine whether a relation exists.

Example 1.

- *Input pair:* ("aspirin", "headache")
- *Candidate relations:* {treats, causes, located_in, part_of}
- *Output:* treats

Example 2.

- *Input pair:* ("Paris", "France")
- *Output:* located_in

Unlike taxonomy discovery, this task captures diverse semantic relations such as causality, functionality, spatial containment, and usage. It is critical for enriching ontologies with expressive relational knowledge.

2 Complexity Score Analysis

The complexity score is computed by 19 metrics grouped into 5 families (graph, coverage, hierarchy, breadth, and dataset). We evaluate the robustness of an ontology complexity score across 180 ontologies using four stress tests, including:

- **Weight perturbation**: $\pm 10\%$ and $\pm 20\%$ for each family (20 variants).
- **Sigmoid sensitivity**: ($a \in [0.3, 0.5]$), ($b \in [5.5, 6.5]$) (24 variants).
- **Normalization alternatives**: min-max and z-score normalizations.
- **Equal-family-weight baseline**: all five families weights set to 0.20.

As a main robustness result, the Table 1a shows that the ranking is nearly unchanged under weight and sigmoid changes, but shifts materially under dataset-dependent normalization (min-max/z-score). This indicates robustness of the aggregation design, with sensitivity concentrated in the

normalization stage. Moreover, the weight Sensitivity analysis (see Table 1b), showed that even $\pm 20\%$ perturbations preserve rank order almost perfectly. Breadth is the most sensitive family (still highly stable), hierarchy the least.

Furthermore, the sigmoid and normalization effect study showed that for different sigmoid parameters, all tested settings produce ($\rho \approx 1.0$), confirming that sigmoid is primarily a monotonic rescaling in this regime. Nevertheless, the normalization method (log vs min-max: $\rho = 0.8557$ and log vs z-score: $\rho = 0.8570$), showed that normalization controls relative spacing across ontologies and can reorder mid-ranked items substantially. This is expected because min-max and z-score are dataset-dependent, whereas $log(1 + x)$ is pointwise and less corpus-sensitive.

The equal-family-weight baseline showed a Spearman correlation of $\rho = 0.9968$ with original weighting, with top-10 overlaps with original ranking. This indicates that the complexity score's discriminative structure is driven more by metric values than by fine-tuned family priors.

The rank stability analysis further supports the robustness of the proposed complexity score. Empirical evidence from the top-50 ontology rankings shows that under moderate weight perturbations (e.g., Graph metrics +20% or Coverage metrics +20%), the highest-ranked ontologies remain largely unchanged, with only minor local swaps (e.g., between SUMO [28] and ChEBI [27] ontologies). Similarly, under a uniform weighting configuration, the top-10 rankings are fully preserved, indicating that the overall ordering is not sensitive to the specific weighting scheme. In contrast, alternative normalization strategies introduce more noticeable shifts: for instance, under min–max normalization, MatOnto [10] moves from rank 32 to 7, CSO [32] from 39 to 6, DBpedia [5] from 33 to 8, and PATO [20] drops from 7 to 18. These results suggest that while most perturbations preserve the relative ranking structure, the choice of normalization has a stronger impact on cross-ontology comparability.

In conclusion, the complexity score is highly robust to both family-weight perturbations (min $\rho = 0.9990$) and sigmoid parameter changes ($\rho \approx 1.0$ across all settings). Equal-family weighting yields near-identical rankings ($\rho = 0.9968$; top-10 overlap 100%), indicating limited dependence on manually chosen family priors. In contrast, the normalization strategy is the dominant sensitivity factor (log vs min-max $\rho = 0.8557$; log vs z-score $\rho = 0.8570$), suggesting that normalization should be fixed a priori and justified explicitly. Overall, the proposed complexity scoring framework is stable under parametric perturbations, while cross-dataset comparability depends primarily on normalization choice.

3 LLMs Contamination Analysis

The strong performance observed in high-resource domains such as *Geography*, *Finance*, and *Units & Measurements* suggests a higher likelihood of dataset contamination, as these domains are more likely to have been encountered by models during pre-training.

Contamination Analysis Dataset. We prepared the contamination analysis dataset by extracting structured knowledge triples from the ontologies across domains of interest. For each ontology, we loaded its extracted representation. We systematically parsed three types of semantic relations: taxonomic relations (*subClassOf* hierarchies), non-taxonomic relations (explicit relational links between entities), and term typings (associations between entities and their types). Each extracted pair was standardized into a uniform schema containing the domain, head entity, relation type, and tail entity. After aggregating results from all ontologies, we merged them into a single dataset, removed duplicates to ensure uniqueness, and used the final collection for downstream contamination analysis.

The extracted dataset contains a total of **1,191** unique ontology-derived relation pairs distributed unevenly across four domains. The majority of the data comes from *Units & Measurements* (901 pairs), followed by *Finance* (283 pairs), while *Geography* (7 pairs) is significantly underrepresented. This imbalance indicates a strong skew toward high-resource structured domains, which may influence downstream analysis and bias contamination estimates toward domains that are more densely populated in existing knowledge sources.

We conducted the contamination analysis by evaluating how well large language models assign probability to ontology-derived relational statements constructed from structured triples. Each triple (h, r, t) was converted into a natural language sentence $x = f(h, r, t)$ (e.g., "h is a type of t" or "h

Table 2: Qwen family of LLMs perplexity statistics by ontology.

LLM	Ontology	Size	Mean	Median	STD	Min	Max	<50%	<100%	<200%	>400%
Qwen3-0.6B	OM	801	251.6	157.3	267.9	20.0	2635.4	7.9	29.5	58.6	18.1
Qwen3-0.6B	GeoNames	7	755.4	285.8	1262.5	73.8	3570.0	0.0	28.6	42.9	28.6
Qwen3-0.6B	QUDT	100	1133.6	310.5	2490.0	30.0	14749.4	3.0	16.0	31.0	39.0
Qwen3-0.6B	GoodRelations	283	14073.4	9730.6	13962.1	24.6	92253.2	0.7	2.8	6.4	93.6
Qwen3-4B-IT-2507	OM	801	133.1	81.2	163.1	9.5	1482.5	30.6	57.2	82.4	4.9
Qwen3-4B-IT-2507	QUDT	100	231.0	174.0	207.5	13.9	1060.5	15.0	29.0	58.0	15.0
Qwen3-4B-IT-2507	GeoNames	7	270.9	69.9	508.8	44.4	1419.7	28.6	71.4	85.7	14.3
Qwen3-4B-IT-2507	GoodRelations	283	7373.5	4552.5	7903.9	13.9	52838.7	2.8	5.3	6.7	93.3
Qwen3-8B	GeoNames	7	118.4	87.3	94.7	36.5	307.5	14.3	57.1	85.7	0.0
Qwen3-8B	OM	801	173.5	99.8	282.3	7.8	4247.4	23.0	50.2	77.8	6.7
Qwen3-8B	QUDT	100	345.7	183.5	766.3	19.8	5546.0	12.0	28.0	55.0	19.0
Qwen3-8B	GoodRelations	283	11974.3	7582.5	13205.6	15.9	94490.1	3.2	4.9	6.7	93.3
Qwen3-14B	GeoNames	7	127.6	72.2	95.0	49.8	277.0	14.3	57.1	71.4	0.0
Qwen3-14B	OM	801	166.2	113.4	192.2	14.5	1823.5	16.5	42.3	75.0	6.1
Qwen3-14B	QUDT	100	299.5	227.2	231.2	29.5	988.5	7.0	21.0	47.0	28.0
Qwen3-14B	GoodRelations	283	11244.6	7377.9	12379.9	27.0	88950.3	1.8	4.6	6.4	93.3
Qwen3-Next-80B-A3B-IT	GeoNames	7	124.0	74.8	141.7	33.9	431.4	28.6	71.4	85.7	14.3
Qwen3-Next-80B-A3B-IT	OM	801	128.7	77.8	150.2	10.4	1175.8	32.6	60.9	81.4	4.9
Qwen3-Next-80B-A3B-IT	QUDT	100	177.5	127.0	162.2	16.1	972.2	13.0	38.0	74.0	9.0
Qwen3-Next-80B-A3B-IT	GoodRelations	283	6148.1	2954.5	9586.1	16.5	64235.6	3.5	5.7	6.7	92.9

is an instance of t"), depending on the relation type. These sentences were then used to compute token-level likelihoods under each model using the causal language modeling objective.

Perplexity-Based Contamination Analysis. For analysis, we used a preplexity-based approach [14], where for a given statement $x = (x_1, x_2, \ldots, x_T)$, the model assigns probability: $P(x) = \prod_{t=1}^{T} P(x_t \mid x_{<t})$, where ee compute the average negative log-likelihood per token as a $\mathcal{L}(x) = -\frac{1}{T}\sum_{t=1}^{T} \log P(x_t \mid x_{<t})$, and the perplexity is defined as a $\mathrm{PPL}(x) = \exp\left(\mathcal{L}(x)\right)$. Lower perplexity indicates that the model assigns a higher likelihood to the statement, which we interpret as a potential signal of memorization or dataset contamination from pretraining data. This procedure was applied in batches across all generated statements for multiple models, and results were aggregated across domains and ontologies to compare likelihood behavior across structured knowledge sources.

Results. We conduct the contamination analysis using the Qwen family of models due to their strong and consistent performance across all evaluated OL tasks, as demonstrated in our benchmark results. The contamination analysis, as shown in Table 2, reveals clear domain-dependent differences in model likelihood behavior. Ontologies from *Units & Measurements* (OM [26], QUDT [16]) consistently exhibit substantially lower perplexity across all models, with mean values decreasing from 251.6 to 128.7 (OM) and from 1133.6 to 177.5 (QUDT) as model size increases. These domains also show a higher proportion of low-perplexity samples (e.g., up to 60.9% below 100 for OM), suggesting that their underlying relations are more likely to be memorized or frequently encountered during pretraining. In contrast, the *Finance* domain ontology (GoodRelations [23]) exhibits extremely high perplexity across all models (mean > 6000 even for the largest model), with over 90% of samples exceeding 400 perplexity, indicating that these structured relations are not well captured by pretrained distributions and are unlikely to reflect memorized knowledge. This highlights that the domain "resource richness" alone does not determine contamination; rather, the extent to which knowledge is expressed in natural language corpora plays a critical role. Additionally, model scaling consistently reduces perplexity across all domains, but preserves the relative ordering, suggesting that contamination signals are stable and not solely driven by model capacity. Finally, results for *Geography* domain remain inconclusive due to the very small sample size (n=7), which leads to high variance and unstable estimates.

4 Experimental Resources and Setup

4.1 Selected Datasets Statistics

Table 3 summarizes the statistics of the selected ontologies used for experimentation with OntoLearner. The selected ontologies span diverse domains and vary significantly in size and structural complexity, enabling evaluation across heterogeneous OL scenarios. We used the full ontologies for evaluation, except for ChEBI [27], evaluation was conducted on a 10% subset of the data due to the significantly larger combinatorial search space, while maintaining the same evaluation protocol.

Table 3: Selected ontological datasets ground truth statistics. The *Terms No.* is the number of terms extracted from the given ontology, whereas the *Types No.* is the number of classes, and *Relations No.* is the number of non-is-a relations. For ontologies that do not support the LLMs4OL tasks or do not contain terms or non-taxonomic relations, frequency values are empty.

	Domain	Ontology	Terms No.	Types No.	Relations No.	Term Typing	Taxonomy Discovery	Non-Taxonomic RE
1	Education	DoCO [12]	-	30	2	-	59	7
2	Events	Conference [29]	32	33	1	32	49	3
3	Finance	GoodRelations [23]	46	25	2	46	25	264
4	Food and Beverage	Wine [21]	161	20	-	161	47	-
5	Agriculture	AgrO [4]	71	4,109	67	71	10,931	1,699
6		FoodOn [15]	16	36,799	6	16	76,228	2,072
7	Geography	GeoNames [1]	699	11	-	699	18	-
8	Materials Science and Engineering	MaterialInformation [3]	404	537	2	404	605	30
9		MatOnto [10]	121	841	2	122	1,215	167
10		PeriodicTable [13]	150	-	-	150	-	-
11		MDSOnto [30]	-	1,332	30	-	1,648	259
12	Biology and Life Sciences	GO [2, 11]	-	40,635	1	-	156,430	30
13	Ecology and Environment	ENVO [8, 9]	44	6,531	5	46	16,175	147
14		SWEET [31]	2,095	10,073	3	2,219	16,111	515
15	General Knowledge	CCO [25]	350	1,422	1	362	1,532	21
16		DBpedia [5]	-	767	4	-	799	1,665
17		SchemaOrg [34]	-	910	1	-	1,058	635
18	Industry	AUTO [17]	57	1,105	1	58	2,731	42
19		PTO [24]	3,000	1,001	-	3,000	3,996	-
20	Medicine	DOID [35]	-	11,731	2	-	41,569	25
21		OBI [7]	285	4,934	3	286	11,843	38
22	Units and Measurements	OM [26]	1,686	798	-	1,953	1,124	-
23		QUDT [16]	23	84	1	27	400	12
24	Chemistry	PROCO [33, 22]	14	831	-	14	1,757	-
25		VIBSO [36]	40	470	3	40	599	23
26		ChEBI [27]	-	201,959	-	-	73,996	-

In the architectural design of the Retriever and Retrieval-Augmented Generation (RAG) modules within OntoLearner, constructing an ontology strictly from scratch requires clearly defined domain boundaries. This is essential to prevent the generation of hallucinated classes or properties that do not belong to the target domain. Standard autoregressive large language models (LLMs) inherently reflect biases from their pre-training distributions, which can introduce irrelevant or incorrect concepts. To address this limitation, the system constrains the model's operational space to grounded domain-specific elements—namely Terms, Types, and Relations—extracted and indexed for each ontology. By restricting the generation process to these validated components, OntoLearner functions as a controlled and reliable assistive tool for ontology engineers. This design principle also motivates the modular structure of the system, where each component is responsible for a single, well-defined task.

In practical settings, ontologies often contain a large number of terms and types, along with a smaller but significant set of non-taxonomic relations. The combinatorial space of possible associations among these elements is substantial and can require extensive manual effort over long periods. Specifically, the term typing task (Term $\rightarrow$ Type), for a dataset of size $|\text{Terms}|$, the total complexity is $O(|\text{Terms}| \times |\text{Types}|)$, while per-instance inference is $O(|\text{Types}|)$. The taxonomy discovery task (Child Type $\rightarrow$ Parent Type) involves identifying hierarchical relationships among types, resulting in a search space of $O(|\text{Types}|^2)$. The non-taxonomic relation extraction task (Head Type $\xrightarrow{Relation}$ Tail Type) is even more complex, as it requires identifying meaningful relational triples across types. This leads to a search space of $O(|\text{Types}|^2 \times |\text{Relations}|)$, representing a significantly larger combinatorial space focused on semantic interactions rather than hierarchical structure. In light of these computational challenges, selecting 26 ontologies provides a balanced and practical benchmark for evaluating OntoLearner while keeping the problem tractable.

4.2 How to Run OntoLearner for LLMs4OL Tasks

The Retrieval-Augmented Generation (RAG) module consists of two main components: a retriever and an LLM-based re-ranker. For using such a technique for *Term Typing* task, consistent with the modular design of OntoLearner, Figure 2 illustrates the retriever implementation using `AutoRetrieverLearner`. Moreover, the *Type Taxonomy Discovery* tasks reuse the term typing formulation, with $t_q \in \mathcal{T}$ as the query instead of $\mathcal{L}$. Thus, in both the retriever (see Figure 2 and re-ranker pipeline (see Figure 3), the only change needed is `task = 'taxonomy-discovery'`. However, for the LLM-Augmented retriever, Figure 4 illustrates the workflow, and Figure 5 shows the end-to-end pipeline usage. The OpenAI model is used as an LLM augmenter, but open-source models can also be applied. LLM-augmented learners are encapsulated and can be adapted for other approaches using the same code. Finally, the *Non-Taxonomic Relation Extraction* task follows a similar pattern, where it reuses the taxonomy discovery and term typing formulations with minimal

```python
from ontolearner import AutoRetrieverLearner, MatOnto, evaluation_report
retriever = AutoRetrieverLearner(top_k=15, batch_size=10240)
retriever.load(model_id='Qwen/Qwen3-Embedding-8B')
ontology = MatOnto()
ontology.load()
data = ontology.extract()
task = 'term-typing'
retriever.fit(data, task=task)
predicts = retriever.predict(data, task=task)
truth = retriever.tasks_ground_truth_former(data=data, task=task)
metrics = evaluation_report(y_true=truth, y_pred=predicts, task=task)
print(metrics)
```

Figure 2: Example code for running the `term-typing` task using a retriever-based learner. To apply the retriever to the two other OL tasks, only the task specification in line 7 needs to be changed (`taxonomy-discovery` for taxonomy discovery and `non-taxonomic-re` for non-taxonomic relationship extraction). The script loads a pretrained embedding model, extracts data from the materials ontology, performs retrieval-based inference, and reports evaluation metrics.

```python
from ontolearner import AutoLLMLearner, AutoRetrieverLearner,
                        StandardizedPrompting, MatOnto,  LabelMapper,
                        AutoRAGLearner,  evaluation_report
llm = AutoLLMLearner(prompting=StandardizedPrompting,
                     label_mapper=LabelMapper(),
                     batch_size=64, device='cuda')
retriever = AutoRetrieverLearner(top_k=15, batch_size=10240)
rag = AutoRAGLearner(llm=llm, retriever=retriever)
rag.load(llm_id='Qwen/Qwen3-14B', retriever_id='Qwen/Qwen3-Embedding-8B')
ontology = MatOnto()
ontology.load()
data = ontology.extract()
task = 'term-typing'
rag.fit(data, task=task)
predicts = rag.predict(data, task=task)
truth = rag.tasks_ground_truth_former(data=data, task=task)
metrics = evaluation_report(y_true=truth, y_pred=predicts, task=task)
print(metrics)
```

Figure 3: Example code for running the `term-typing` task using a RAG learner. To apply the retriever to other tasks, only the task specification in line 13 needs to be changed (`taxonomy-discovery` for taxonomy discovery and `non-taxonomic-re` for non-taxonomic relationship extraction). The script initializes the LLM and retriever components, loads pretrained models, extracts data from the materials ontology, performs RAG-based training and inference, and reports evaluation metrics.

modifications. In OntoLearner, either in the retriever (Figure 2) or reranker (Figure 3), it is enabled via `task = 'non-taxonomic-re'`, integrating pair discovery, natural language query generation, relation retrieval, and LLM verification in a unified RAG pipeline.

4.3 Standardized Prompts for LLMs

We employ standardized zero-shot prompts with strict output constraints (yes/no) to ensure comparability across tasks. Prompts were designed to minimize ambiguity and avoid task leakage. The prompts are supported via OntoLearner using `AutoPrompt` module and presented in the Figure 6. We enforce binary outputs to ensure consistency across models and enable reliable and comparable evaluation across tasks.

4.4 Computational Details

All experiments were conducted in a zero-shot setting using pretrained models without task-specific fine-tuning. Unless otherwise specified, default inference parameters were used for all large language models (LLMs).

Hardware and Execution. Experiments were performed on a multi-GPU system equipped with NVIDIA H100 GPUs. Larger LLMs (e.g., 27B–80B parameters) were executed on configurations utilizing two H100 GPUs (95 GB memory each), while smaller models were experimented on a

```python
from ontolearner import LLMAugmenterGenerator, LLMAugmenter,
                        LLMAugmentedRetriever, LLMAugmentedRetrieverLearner,
                        MatOnto, evaluation_report, utils
augmenter = LLMAugmenterGenerator(token='openai-api-key')
ontology = MatOnto()
ontology.load()
data = ontology.extract()
task = 'taxonomy-discovery'
path = f"{task}-{ontology.ontology_id}.json"
augments = {"config": augmenter.get_config(),
            task: augmenter.augment(data, task=task)}
utils.save_json(path, augments)
augmenter = LLMAugmenter(path=path)
base_retriever = LLMAugmentedRetriever()
retriever = LLMAugmentedRetrieverLearner(base_retriever=base_retriever,
                                         top_k=15, batch_size=10240)
retriever.load(model_id='Qwen/Qwen3-Embedding-8B')
retriever.set_augmenter(augmenter=augmenter)
retriever.fit(data, task=task)
predicts = retriever.predict(data, task=task)
truth = retriever.tasks_ground_truth_former(data=data, task=task)
metrics = evaluation_report(y_true=truth, y_pred=predicts, task=task)
print(metrics)
```

Figure 4: Example code illustrating the end-to-end workflow for taxonomy discovery task using an LLM-augmented retriever, including ontology loading, data augmentation, retrieval-based learning, and evaluation. When a task other than `taxonomy-discovery` is specified in line 8 (i.e., `term-typing` or `non-taxonomic-re`), the augmenter defaults to a retriever-based model, as illustrated in Figure 2.

```python
from ontolearner import LLMAugmentedRetriever, LLMAugmentedRetrieverLearner,
                        LLMAugmentedRAGLearner, MatOnto,  LabelMapper,
                        StandardizedPrompting,  evaluation_report
llm = AutoLLMLearner(prompting=StandardizedPrompting,
                     label_mapper=LabelMapper(),
                     batch_size=64,
                     device='cuda')
base_retriever=LLMAugmentedRetriever()
retriever = LLMAugmentedRetrieverLearner(base_retriever=base_retriever,
                                         top_k=15, batch_size=10240)
rag = LLMAugmentedRAGLearner(llm=llm, retriever=retriever)
rag.load(llm_id='Qwen/Qwen3-14B', retriever_id='Qwen/Qwen3-Embedding-8B')
ontology = MatOnto()
ontology.load()
data = ontology.extract()
task = 'taxonomy-discovery'
augmenter = LLMAugmenter(path=f"{task}/{ontology.ontology_id}.json")
rag.fit(data, task=task)
predicts = rag.predict(data, task=task)
truth = rag.tasks_ground_truth_former(data=data, task=task)
metrics = evaluation_report(y_true=truth, y_pred=predicts, task=task)
print(metrics)
```

Figure 5: Example code for running the RAG learner for `taxonomy-discovery` using an LLM-augmented retriever. The augmented retriever is activated only when `taxonomy-discovery` is specified; for other tasks (i.e., `term-typing` or `non-taxonomic-re`), the retriever falls back to standard RAG behavior without augmentation, as illustrated in Figure 3.

single H100 GPU. Multiple runs were executed concurrently across available GPUs. This parallel runs enabled large-scale evaluation across all ontologies and tasks within a reasonable time frame, while maintaining consistent experimental conditions.

Batching and Inference. Batch sizes were adjusted depending on the model size and task requirements, as reflected in the code examples provided in Figure 5. Retrieval and reranking components were executed using pretrained embedding and LLM models without additional optimization or pruning. All tasks—term typing, taxonomy discovery, and non-taxonomic relation extraction—were evaluated using grouped metrics (precision, recall, and F1-score) computed over the full set of ontology datasets.

Run Time. We evaluated the robustness and computational scalability of OntoLearner across all datasets and 12 LLMs. All models achieved a 100% completion rate (**756 runs**), demonstrating

Term Typing Prompt

You are performing term typing.

Determine whether the given term is a clear and unambiguous instance of the specified high-level type.

Rules:
- Answer ”yes” only if the term commonly and directly belongs to the type.
- Answer ”no” if the term does not belong to the type, is ambiguous, or only weakly related.
- Use the most common meaning of the term.
- Do not explain your answer.

Term: {term}
Type: {type}
Answer (yes or no):

Taxonomy Discovery Prompt

You are identifying taxonomic (is-a) relationships.

Determine whether the first concept is a superclass (direct or indirect) of the second concept in a standard conceptual hierarchy.

Rules:
- A superclass means the second concept is a type or instance of the first.
- Answer ”yes” only if the relationship is a true is-a relationship.
- Answer ”no” for part-of, related-to, or associative relationships.
- Use widely accepted general knowledge.
- Do not explain your answer.

Parent: {parent}
Child: {child}
Answer (yes or no):

Non-Taxonomic Relation Extraction Prompt

You are identifying non-taxonomic conceptual relationships.

Determine whether the specified relation typically holds between the given conceptual types.

Rules:
- Answer ”yes” only if the relation commonly and meaningfully applies.
- Answer ”no” if the relation is rare, indirect, or context-dependent.
- Do not infer relations that require specific situations.
- Use widely accepted general knowledge.
- Do not explain your answer.

Head type: {head}
Tail type: {tail}
Relation: {relation}
Answer (yes or no):

Figure 6: Standardized prompt templates for LLM-based reranker of term typing, taxonomy discovery, and non-taxonomic relation extraction tasks.

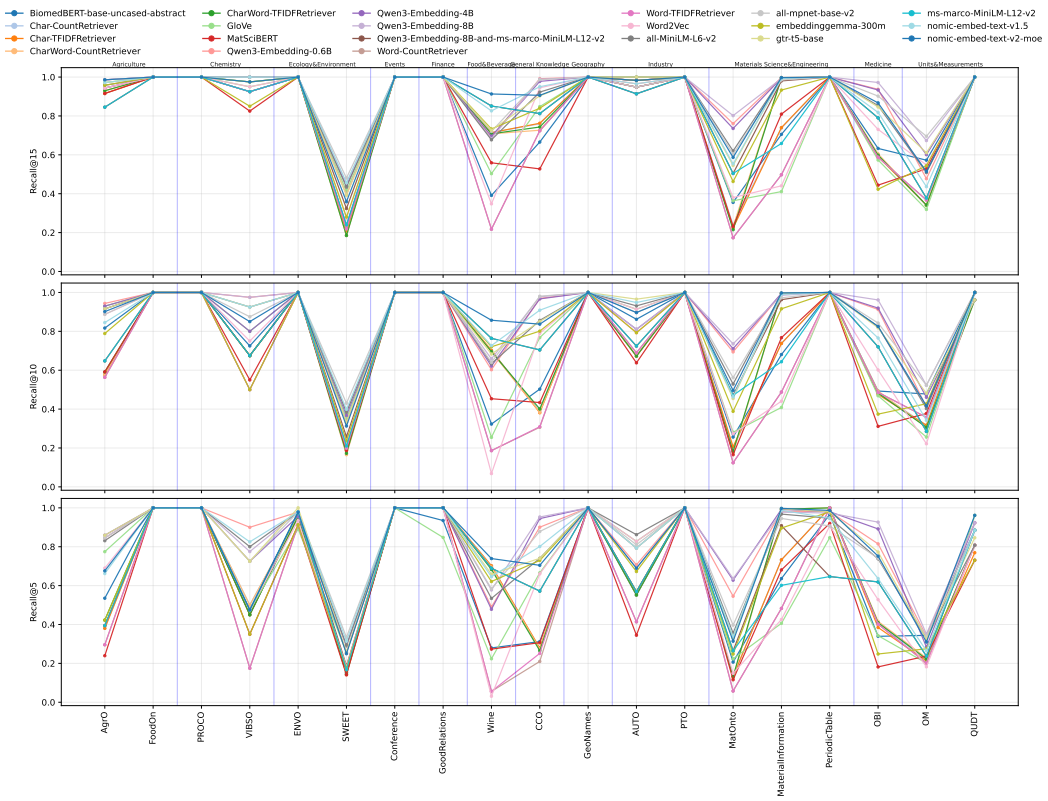


Figure 7: Performance comparison of retrieval-based methods on the **term typing** task.

stable and failure-free execution across tasks and domains. Each LLM was evaluated on 63 tasks, corresponding to combinations of OL tasks and datasets. The average response time per run was approximately 21,566 seconds (∼6 hours), resulting in a total runtime of approximately **378** hours per LLM and an aggregate runtime of **4,536** hours across all models. These experiments were executed in parallel across multiple GPUs, enabling large-scale evaluation within practical time constraints while maintaining consistent experimental conditions. Substantially higher runtimes were observed for large-scale ontologies such as ChEBI, where the average runtime reached approximately 387,398 seconds per run. This highlights the dominant role of ontology size, particularly the number of types, in determining computational cost. In particular, tasks such as taxonomy discovery exhibit quadratic complexity with respect to the number of types, leading to significantly larger search spaces in high-dimensional ontologies. Overall, these results demonstrate that OntoLearner scales reliably across diverse models and domains, while also revealing that computational efficiency is primarily constrained by the combinatorial structure of OL tasks. This suggests that future optimizations should focus on reducing the effective search space, for example, through candidate pruning or adaptive retrieval strategies.

All experiments can be reproduced on smaller subsets using the provided modular pipeline.

5 Detailed Results

5.1 Retrievers

The Figures 7–9 present the performance of retrieval-based approaches across the three OL tasks: term typing, taxonomy discovery, and non-taxonomic relationship extraction. We report grouped recall to assess how effectively each retriever supports downstream OL objectives. Overall, the results reveal task-dependent performance variations, highlighting the importance of retrieval quality in enabling accurate concept classification, hierarchical structuring, and relational inference.

Figure 7 shows that dense embedding-based retrievers consistently outperform lexical baselines across most domains, often by a substantial margin in grouped recall. This indicates that semantic similarity captured by dense representations is critical for accurate term typing, particularly in domains with high lexical variability or synonymy. In contrast, lexical methods struggle when surface forms diverge from canonical ontology labels.

Moreover, Figure 8 further demonstrates that LLM-Augmented retrieval yields consistent improvements over standard retrieval approaches in taxonomy discovery. The gains are especially pronounced in larger and more complex ontologies, suggesting that augmentation helps bridge the gap between surface-level similarity and deeper hierarchical reasoning. This highlights the importance of incorporating contextual or generative signals when modeling subsumption relationships.

Finally, Figure 9 reveals substantially higher variability in performance across domains for non-taxonomic relation extraction. This reflects the inherently diverse and context-dependent nature of such relations, which often require richer semantic understanding beyond similarity or hierarchy. As a result, retrieval performance in this setting is more sensitive to domain characteristics and relation distributions.

5.2 Reranking for the RAG Pipeline

The results in Figures 10–12 highlight clear trade-offs between precision and recall in different LLMs as a reranker in the RAG pipeline across OL tasks. While some LLMs achieve high precision by aggressively filtering candidates, this often comes at the cost of reduced recall, particularly in tasks with large candidate spaces such as taxonomy discovery. In contrast, recall shows that LLMs provide broader coverage but may introduce more false positives, impacting the overall F1-score. The balance between these behaviors varies across tasks: term typing benefits from higher precision due to well-defined class boundaries, whereas taxonomy discovery and non-taxonomic relation extraction benefit from higher recall to capture diverse valid relationships. Overall, these results emphasize the importance of selecting LLMs that align with task-specific requirements, as no single approach uniformly performs well across all OL tasks.

References

[1] Geonames ontology (geonames). `https://www.geonames.org/ontology`, 2022. RDF ontology. Licensed under Creative Commons 3.0. Last updated: 2022-01-30.

[2] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[3] Toshihiro Ashino. Materials ontology: An infrastructure for exchanging materials information and knowledge. *Data Science Journal*, 9:54–61, 2010.

[4] Céline Aubert, Marie-Angélique Laporte, Krishna Udaiwal, Pier Luigi Buttigieg, Chris Mungall, Elizabeth Arnaud, and Charles Tapley Hoyt. Agriculturalsemantics/agro: November 2022 release, November 2022.

[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pages 722–735. Springer, 2007.

[6] Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. Llms4ol: Large language models for ontology learning. In *International Semantic Web Conference*, pages 408–427. Springer, 2023.

[7] Anita Bandrowski, Ryan Brinkman, Mathias Brochhausen, Matthew H Brush, Bill Bug, Marcus C Chibucos, Kevin Clancy, Mélanie Courtot, Dirk Derom, Michel Dumontier, et al. The ontology for biomedical investigations. *PloS one*, 11(4):e0154556, 2016.

[8] Pier Luigi Buttigieg, Norman Morrison, Barry Smith, Christopher J Mungall, Suzanna E Lewis, and Envo Consortium. The environment ontology: contextualising biological and biomedical entities. *Journal of biomedical semantics*, 4(1):43, 2013.

[9] Pier Luigi Buttigieg, Evangelos Pafilis, Suzanna E Lewis, Mark P Schildhauer, Ramona L Walls, and Christopher J Mungall. The environment ontology in 2016: bridging domains with increased scope, semantic density, and interoperation. *Journal of biomedical semantics*, 7(1):57, 2016.

[10] Kwok Cheung, John Drennan, and Jane Hunter. Towards an ontology for data-driven discovery of new materials. In *AAAI Spring Symposium: Semantic Scientific Knowledge Integration*, pages 9–14, 2008.

[11] The Gene Ontology Consortium, Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, David P Hill, Raymond Lee, Huaiyu Mi, Sierra Moxon, Christopher J Mungall, Anushya Muruganugan, Tremayne Mushayahama, Paul W Sternberg, Paul D Thomas, Kimberly Van Auken, Jolene Ramsey, Deborah A Siegele, Rex L Chisholm, Petra Fey, Maria Cristina Aspromonte, Maria Victoria Nugnes, Federica Quaglia, Silvio Tosatto, Michelle Giglio, Suvarna Nadendla, Giulia Antonazzo, Helen Attrill, Gil dos Santos, Steven Marygold, Victor Strelets, Christopher J Tabone, Jim Thurmond, Pinglei Zhou, Saadullah H Ahmed, Praoparn Asanitthong, Diana Luna Buitrago, Meltem N Erdol, Matthew C Gage, Mohamed Ali Kadhum, Kan Yan Chloe Li, Miao Long, Aleksandra Michalak, Angeline Pesala, Armalya Pritazahra, Shirin C C Saverimuttu, Renzhi Su, Kate E Thurlow, Ruth C Lovering, Colin Logie, Snezhana Oliferenko, Judith Blake, Karen Christie, Lori Corbani, Mary E Dolan, Harold J Drabkin, David P Hill, Li Ni, Dmitry Sitnikov, Cynthia Smith, Alayne Cuzick, James Seager, Laurel Cooper, Justin Elser, Pankaj Jaiswal, Parul Gupta, Pankaj Jaiswal, Sushma Naithani, Manuel Lera-Ramirez, Kim Rutherford, Valerie Wood, Jeffrey L De Pons, Melinda R Dwinell, G Thomas Hayman, Mary L Kaldunski, Anne E Kwitek, Stanley J F Laulederkind, Marek A Tutaj, Mahima Vedi, Shur-Jen Wang, Peter D'Eustachio, Lucila Aimo, Kristian Axelsen, Alan Bridge, Nevila Hyka-Nouspikel, Anne Morgat, Suzi A Aleksander, J Michael Cherry, Stacia R Engel, Kalpana Karra, Stuart R Miyasato, Robert S Nash, Marek S Skrzypek, Shuai Weng, Edith D Wong, Erika Bakker, Tanya Z Berardini, Leonore Reiser, Andrea Auchincloss, Kristian Axelsen, Ghislaine Argoud-Puy, Marie-Claude Blatter, Emmanuel Boutet, Lionel Breuza, Alan Bridge, Cristina Casals-Casas, Elisabeth Coudert, Anne Estreicher, Maria Livia Famiglietti, Marc Feuermann, Arnaud Gos, Nadine Gruaz-Gumowski, Chantal Hulo, Nevila Hyka-Nouspikel, Florence Jungo, Philippe Le Mercier, Damien Lieberherr, Patrick Masson, Anne Morgat, Ivo Pedruzzi, Lucille Pourcel, Sylvain Poux, Catherine Rivoire, Shyamala Sundaram, Alex Bateman, Emily Bowler-Barnett, Hema Bye-A-Jee, Paul Denny, Alexandr Ignatchenko, Rizwan Ishtiaq, Antonia Lock, Yvonne Lussi, Michele Magrane, Maria J Martin, Sandra Orchard, Pedro Raposo, Elena Speretta, Nidhi Tyagi, Kate Warner, Rossana Zaru, Alexander D Diehl, Raymond Lee, Juancarlos Chan, Stavros Diamantakis, Daniela Raciti, Magdalena Zarowiecki, Malcolm Fisher, Christina James-Zorn, Virgilio Ponferrada, Aaron Zorn, Sridhar Ramachandran, Leyla Ruzicka, and Monte Westerfield. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 03 2023.

[12] Alexandru Constantin, Silvio Peroni, Steve Pettifer, David Shotton, and Fabio Vitali. The document components ontology (doco). *Semantic web*, 7(2):167–181, 2016.

[13] Michael Cook. Periodic table of the elements ontology (periodictable). `https://www.daml.org/2003/01/periodictable/`, February 2004. OWL ontology. Last updated: 2004-02-05.

[14] Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12039–12050, 2024.

[15] Damion M Dooley, Emma J Griffiths, Gurinder S Gosal, Pier L Buttigieg, Robert Hoehndorf, Matthew C Lange, Lynn M Schriml, Fiona SL Brinkman, and William WL Hsiao. Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration. *npj Science of Food*, 2(1):23, 2018.

[16] FAIRsharing.org. QUDT: Quantities, Units, Dimensions and Types. Online, 2025. Accessed: 2026-01-06; Last edited: 2025-11-03.

[17] Michael Feld and Christian Müller. The automotive ontology: managing knowledge inside the vehicle and sharing it between cars. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 79–86, 2011.

[18] Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. Llms4ol 2024 overview: The 1st large language models for ontology learning challenge. *arXiv preprint arXiv:2409.10146*, 2024.

[19] Hamed Babaei Giglou, Jennifer D'Souza, Nandana Mihindukulasooriya, and Sören Auer. Llms4ol 2025 overview: The 2nd large language models for ontology learning challenge. In *Open Conference Proceedings*, volume 6, 2025.

[20] Georgios V Gkoutos, Paul N Schofield, and Robert Hoehndorf. The anatomy of phenotype ontologies: principles, properties and applications. *Briefings in bioinformatics*, 19(5):1008–1021, 2018.

[21] João Graça, Márcio Mourao, Orlando Anunciação, Pedro Monteiro, H Sofia Pinto, and Virgílio Loureiro. Ontology building process: the wine domain. In *Proc. of the 5th Conf. of EFITA*, 2005.

[22] Oliver He and Wes Schafer. Extending the allotrope framework: An ontological representation and analysis of process chemistry. 2020 Fall Allotrope Connect Virtual Conference (Oral presentation), September 2020. Virtual conference, September 30, 2020.

[23] Martin Hepp. Goodrelations: An ontology for describing products and services offers on the web. In *International conference on knowledge engineering and knowledge management*, pages 329–346. Springer, 2008.

[24] Martin Hepp, Katharina Siorpaes, and Daniel Bachlechner. Harvesting wiki consensus: Using wikipedia entries as vocabulary for knowledge management. *IEEE Internet Computing*, 11(5):54–65, 2007.

[25] Mark Jensen, Giacomo De Colle, Sean Kindya, Cameron More, Alexander P Cox, and John Beverley. The common core ontologies. *arXiv preprint arXiv:2404.17758*, 2024.

[26] Jan Martin Keil and Sirko Schindler. Comparison and evaluation of ontologies for units of measurement. *Semantic Web*, 10(1):33–51, 2018.

[27] Adnan Malik, Muhammad Arsalan, Carlos Moreno, Juan Mosquera, Eloy Félix, Tevfik Kizilören, Venkatesh Muthukrishnan, Barbara Zdrazil, Andrew R Leach, and Noel M O'Boyle. Chebi: re-engineered for a sustainable future. *Nucleic Acids Research*, page gkaf1271, 2025.

[28] Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9, 2001.

[29] Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, and Aldo Gangemi. Semantic web conference ontology - a refactoring solution. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenić, Sören Auer, and Christoph Lange, editors, *The Semantic Web*, pages 84–87, Cham, 2016. Springer International Publishing.

[30] Balashanmuga Priyan Rajamohan, Alexander C Harding Bradley, Van D Tran, Jonathan E Gordon, Hayden W Caldwell, Redad Mehdi, Gabriel Ponon, Quynh D Tran, Ozan Dernek, Jarod Kaltenbaugh, et al. Materials data science ontology (mds-onto): Unifying domain knowledge in materials and applied data science. *Scientific Data*, 12(1):628, 2025.

[31] Rob Raskin and Michael Pan. Semantic web for earth and environmental terminology (sweet). In *Proc. of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, volume 25, 2003.

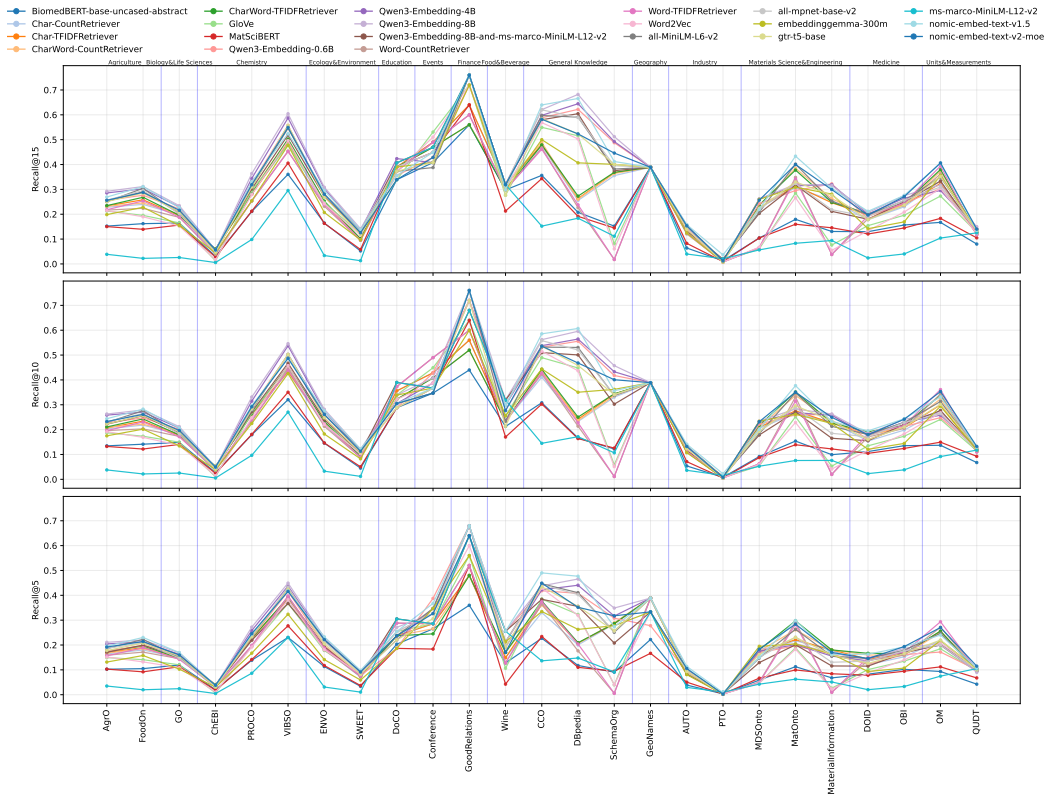
[32] Angelo A Salatino, Thiviyan Thanapalasingam, Andrea Mannocci, Francesco Osborne, and Enrico Motta. The computer science ontology: a large-scale taxonomy of research areas. In *International Semantic Web Conference*, pages 187–205. Springer, 2018.

[33] Wes Schafer, Oliver He, Anna L. Dunn, and Zachary E. X. Dance. Ontology for process chemistry – giving context to instrument data structured by the allotrope data model. Allotrope Connect Virtual Conference, April 2021. Virtual conference, April 19–26, 2021.

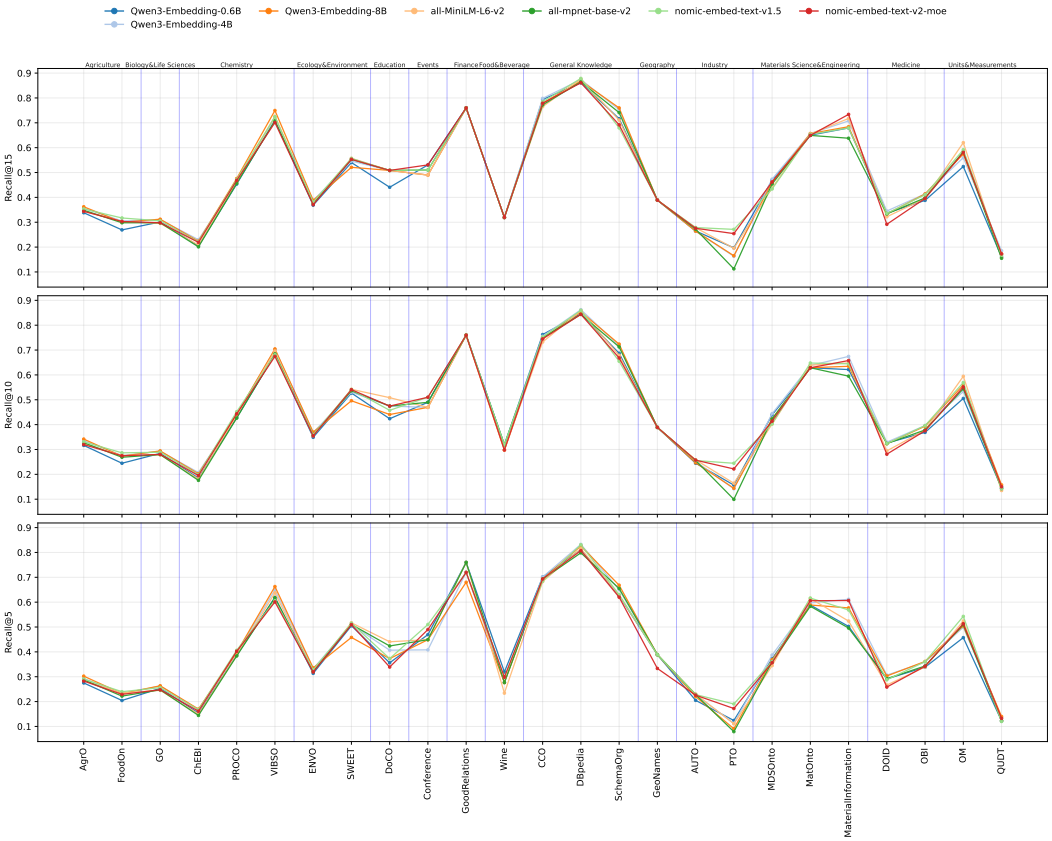
[34] Schema.org Community Group. Schema.org. `https://schema.org/`, 2011.

[35] Lynn M Schriml, Elvira Mitraka, James Munro, Becky Tauber, Mike Schor, Lance Nickle, Victor Felix, Linda Jeng, Cynthia Bearer, Richard Lichenstein, Katharine Bisordi, Nicole Campion, Brooke Hyman, David Kurland, Connor Patrick Oates, Siobhan Kibbey, Poorna Sreekumar, Chris Le, Michelle Giglio, and Carol Greene. Human disease ontology 2018 update: classification, content and workflow expansion. *Nucleic Acids Research*, 47(D1):D955–D962, 11 2018.

[36] Philip Strömert, Giacomo Lanza, Johannes Hunold, Steffen Neumann, et al. The Vibrational Spectroscopy Ontology (VIBSO), 2023.



(a) Traditional Retriever Approach



(b) LLM-Augmented Retriever Approach

Figure 8: Performance comparison of retrieval-based methods on the **taxonomy discovery** task.

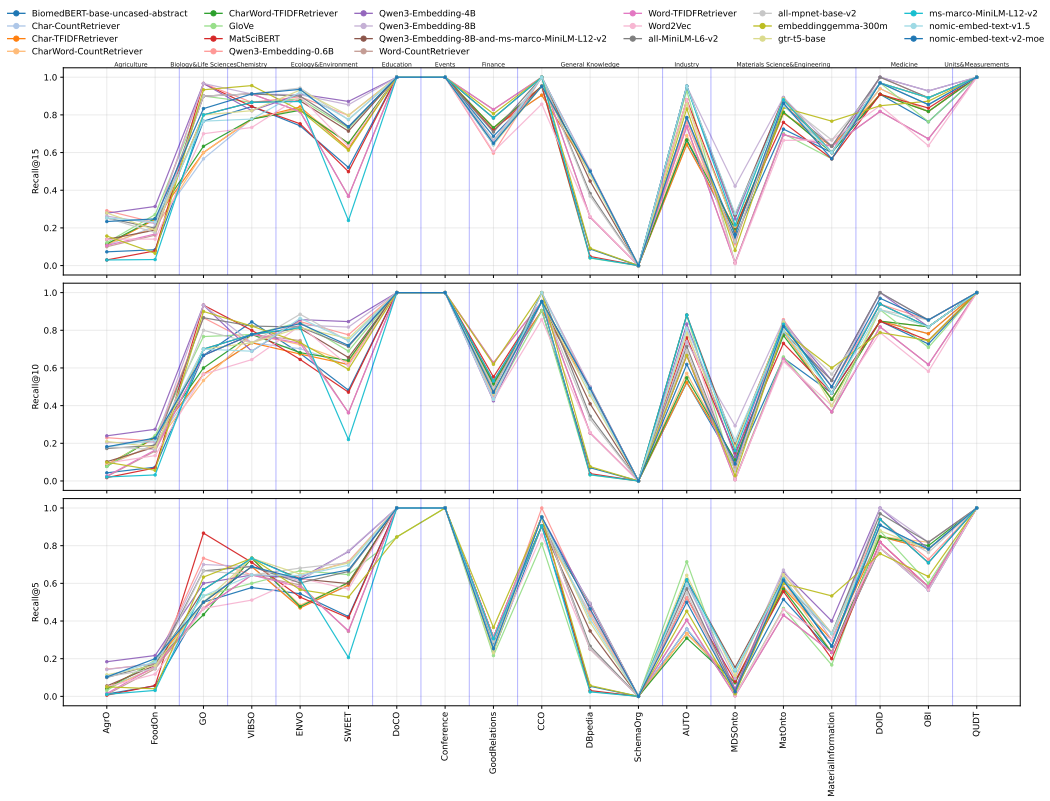


Figure 9: Performance comparison of retrieval-based methods on the **non-taxonomic relationship extraction** task.

Figure 10: Comparison of reranking methods across the three OL tasks—term typing, taxonomy discovery, and non-taxonomic relationship extraction—measured using grouped **F1-score**. The figure summarizes the overall balance between precision and recall achieved by each reranker across tasks and domains.

Figure 11: Comparison of reranking methods across the three OL tasks using grouped **precision**. The results highlight the ability of each reranker to return highly relevant candidates while minimizing false positives in different ontology learning settings.

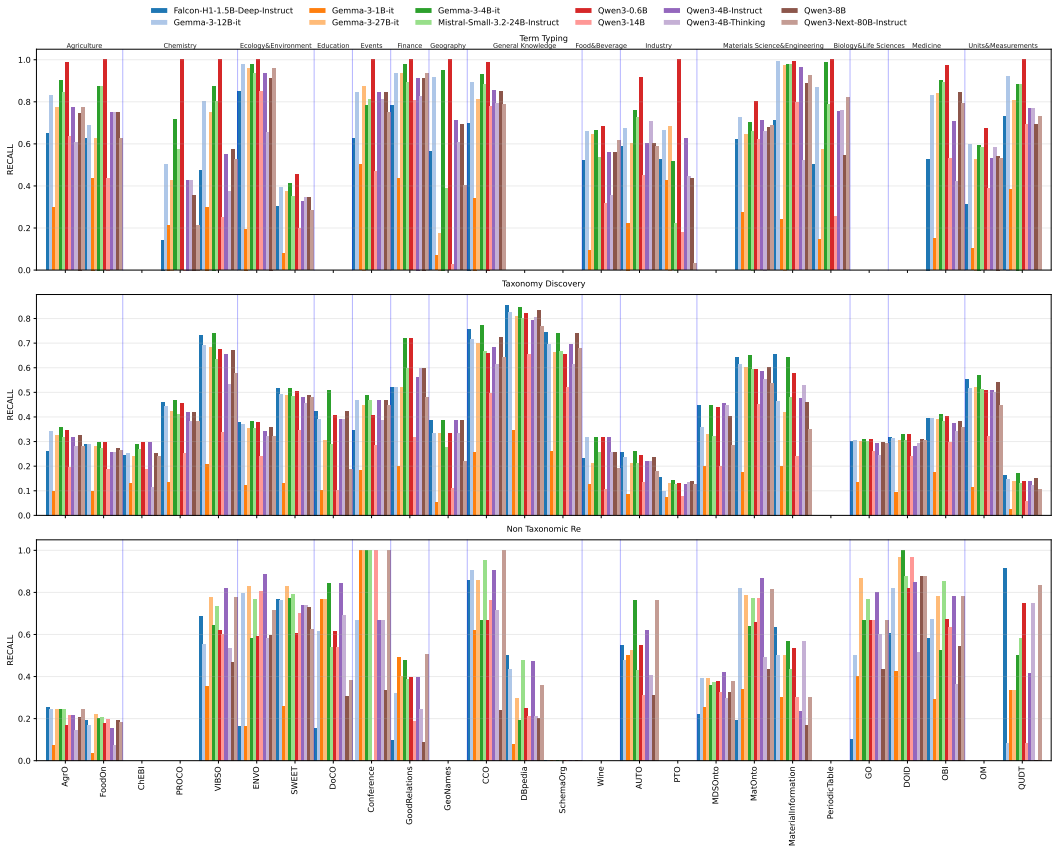


Figure 12: Comparison of reranking methods across the three OL tasks using grouped **recall**. The figure reflects how effectively each approach retrieves relevant candidates, emphasizing coverage across term typing, taxonomy discovery, and relation extraction tasks.