LLMs4OL 2025: The 2nd Large Language Models for Ontology Learning Challenge at the 24th ISWC

LLMs4OL 2025 Task Participant Short Papers

https://doi.org/10.52825/ocp.v6i.2896

Published: 01 Oct. 2025

CUET_Zenith at LLMs4OL 2025 Task C: Hybrid Embedding-LLM Architectures for Taxonomy Discovery

Rehenuma Ilman[1], Mehreen Rahman[1,*], and Samia Rahman[1]

[1]Chittagong University of Engineering and Technology, Chittagong, Bangladesh

*Correspondence: Mehreen Rahman, u2004033@student.cuet.ac.bd

Abstract. Taxonomy discovery, the identification of hierarchical relationships within ontological structures, constitutes a foundational challenge in ontology learning. Our submission to the LLMs4OL 2025 challenge, employing hybrid architectures to address this task across both biomedical (Subtask C1: OBI) and general-purpose (Subtask C5: SchemaOrg) knowledge domains. For C1, we have integrated semantic clustering of Sentence-BERT embeddings with few-shot prompting using Qwen-3 (14B), enabling domain-specific hierarchy induction without task-specific fine-tuning. For C5, we have introduced a cascaded validation framework, harmonizing deep semantic representations from sentence transformer all-mpnet-base-v2, ensemble classification via XGBoost, and a hierarchical LLM-based reasoning pipeline utilizing TinyLlama and GPT-4o. To address inherent class imbalances, we have employed SMOTE-based augmentation and gated inference thresholds. Empirical results demonstrate that our hybrid methodology achieves competitive performance, confirming that the judicious integration of classical machine learning with large language models yields efficient and scalable solutions for ontology structure induction. Code implementations are publicly available.

Keywords: Ontology Learning, Taxonomy Discovery, Large Language Models, Hybrid Architectures, Biomedical Ontologies .

1. Introduction

Ontology Learning (OL) constructs the foundational aspect for structuring the raw and textual data in the field of artificial intelligence. The first edition of the LLMs4OL challenge, introduced at ISWC 2024 [1], provided a structured and competitive framework for assessing how effectively large language models can be applied to core ontology learning tasks. As a logical extension of the groundwork laid in previous years, the 2nd LLMs4OL challenge @ ISWC 2025 [2] was organized with the goal of expanding our knowledge of the potential of large language models for ontology learning[1] Our team CUET_Zenith has participated in Task C: Taxonomy Discovery, which focuses on identifying hierarchical (is-a) relationships between ontological type pairs. The objective of this task is to identify taxonomic links within given datasets. Among the given datasets

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

we have chosen the SubTask C1: OBI that focuses on biomedical investigation types using the Ontology for Biomedical Investigations (OBI) and SubTask C5: SchemaOrg that focuses on general-purpose web knowledge concepts using the widely adopted SchemaOrg vocabulary.

For Subtask C1, We propose a hybrid approach that integrates semantic similarity-based clustering using sentence transformer embeddings with few-shot prompting from large language models (LLMs). Among the various approaches evaluated, the combination of the all-MiniLM-L6-v2 sentence embedding model [3] and AgglomerativeClustering [4] yielded strong result for identifying taxonomic relationships. Additionally, the use of Qwen3-14B [5] through the unsloth fine-tuning framework [6] significantly enhanced performance in zero-shot prediction scenarios.Implementation of this architecture can be found in our GitHub repository[2].

For Subtask C5, we propose a hybrid framework for Schema.org taxonomy discovery that synergistically integrates semantic embedding techniques with cascaded language model validation. The methodology employs:

- Semantic Representation: all-mpnet-base-v2 embeddings capturing ontological semantics [7].
- Feature Synthesis: Engineered vectors combining embedding operations and lexical patterns [8].
- Ensemble Classification: XGBoost architecture with imbalance compensation [9].
- Cascaded Verification: Two-tier LLM validation combining binary assessment and probabilistic scoring [10],[11].

This architecture balances precision with computational efficiency through strategic validation gating. Implementation available in our github repository [12] .

2. Related Work

Ontology Learning (OL) is a foundational challenge in artificial intelligence and knowledge engineering, aiming to automate the acquisition of structured knowledge from unstructured data. Early OL approaches relied heavily on expert-driven methods or statistical techniques like lexico-syntactic pattern mining and clustering [13], [14]. As demands for scalability and domain adaptation grew, machine learning techniques such as TF-IDF classifiers and hierarchical clustering were adopted to automate taxonomy induction and term typing [15].

Recent advances in language modeling have introduced new paradigms for OL through Large Language Models (LLMs). The LLMs4OL initiative [1] explores this potential across multiple subtasks such as term typing, taxonomy discovery, and relation extraction. These tasks have seen growing adoption of hybrid architectures that combine embedding-based retrieval with transformer-based inference.

Prior studies have demonstrated the effectiveness of LLMs in taxonomy induction, often leveraging hybrid architectures that balance classical methods and generative reasoning. The SKH-NLP system at LLMs4OL 2024 [16] explored BERT and LLaMA-3 for binary taxonomic relation classification on the GeoNames ontology, highlighting the comparative strengths of fine-tuned and zero-shot LLM performance. Similarly, participants in Subtask B [17], [18] have leveraged combinations of classical classifiers and generative prompting to approximate is-a hierarchies across domains. These

[2] https://github.com/Mehreen1103/LLMs4OL-2025_Task-C

approaches have informed our model design by validating the complementary roles of embeddings, lexical signals, and language model reasoning.

Building on these insights, our submission addresses Subtask C5 (Schema.org) and Subtask C1 (OBI). For C1, a biomedical ontology, domain-aligned clustering and prompt engineering were used to adapt sentence embeddings to medical hierarchies. For C5, which emphasizes general-purpose web concepts, we designed a hybrid system that blends sentence transformer-based similarity with XGBoost classification, followed by cascaded LLM-based filtering using TinyLlama and GPT-4o. This combination balances precision and coverage across domain-diverse ontological hierarchies.

3. Dataset

3.1 Dataset SubTask-C1

For subtask C1 the training dataset contains a json file carrying is-a relationships and a text file stating the unique ontology one per line. Each entry in the training pair includes an ID, parent, and child field. The test data set only contains the list of the type of ontology. The table showing the statistics of the dataset is given in the table 1.

The JSON file containing labeled parent child is-a pairs in the following format:

```
[
  {
   "ID": "TR_bb9941a6",
    "parent": "hemoglobin assay",
    "child": "cooximitery arterial blood hemoglobin assay"
  },
  ...
]
```

Table 1. *Subtask C1 Dataset Statistics*

Dataset	File Name	Number of Entries
Training	`obi_train_pairs.json`	8,249 pairs
	`obi_train_types.txt`	4,237 types
Test	`obi_test_types.txt`	2,821 types

3.2 Dataset for SubTask-C5

For subtask C5, the training dataset includes a JSON file with labeled is-a relationships and a text file listing unique ontology types. Each training entry specifies an ID, parent, and child type. The test dataset provides a list of unseen ontology types for which parent types must be predicted. The dataset is publicly available [19]. Dataset statistics are summarized in Table 2.

The training JSON file contains is-a pairs in the following format:

```
[
  {
    "ID": "TR_56ac8cd6",
    "parent": "Enumeration",
    "child": "WarrantyScope"
```

```
  },
  ...
]
```

Table 2. *Subtask C5 Dataset Statistics*

Dataset	File Name	Number of Entries
Training	`schemaorg_train_pairs.json`	723 pairs
	`schemaorg_train_types.txt`	692 types
Test	`schemaorg_test_types.txt`	359 types

4. Methodology

4.1 Data Augmentation

The training corpus for the BestTaxonomyClassifier comprises labeled (parent, child) pairs from the Schema.org ontology, each denoting a valid is-a subclass relationship. To reframe the task as binary classification, an equal number of negative instances were systematically constructed using domain-aware heuristics.

Two complementary strategies were adopted to synthesize negative examples:

1. **Reversed Pairs:** Approximately one-third of the negatives were created by inverting the direction of positive pairs (e.g., `Movie` $\rightarrow$ CreativeWork becomes CreativeWork $\rightarrow$ Movie), thereby violating the inherent asymmetry of subclass relations.
2. **Manipulated Pairs:** The remaining two-thirds were generated by replacing the original parent with a randomly sampled alternative from the type vocabulary. To preserve semantic invalidity, candidate substitutions that were valid, substring-contained, or lexically similar were excluded.

Each pair was encoded using contextual sentence embeddings (all-mpnet-base-v2)[3], enriched with lexical overlap and pairwise similarity features. To address minor class imbalance post-stratified split, the Synthetic Minority Over-sampling Technique (SMOTE) was employed. A quantitative summary of the augmentation and balancing process is presented in Table 3.

Table 3. *Data Augmentation and Class Balancing Summary*

Component	Count
Positive Pairs (Original)	723
Negative Pairs (Total)	723
– Reversed	241
– Manipulated	482
Training Set (Before SMOTE)	1446
Training Set (After SMOTE)	2875
– Positive Samples	1425
– Negative Samples	1450

4.2 Overview of the proposed model

4.2.1 Proposed model for subtask C1

Our proposed hybrid approach combines sentence embedding, clustering, and few-shot prompting using large language models. This three-stage pipeline integrates both

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

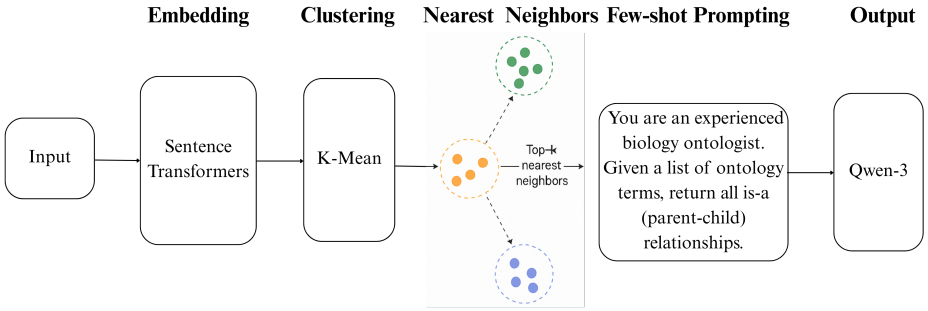

Figure 1. *Overview of the proposed model*

semantic grouping and hierarchical reasoning to enable effective extraction of parent-child is-a relationships.

Since the dataset is large and contains a vast number of ontology types, we first encode each term into a high-dimensional vector representation using Sentence-BERT.

Step 2: Agglomerative Clustering

In the next stage, we perform agglomerative hierarchical clustering to organize semantically similar ontology types into clusters. In this process, we define the affinity metric as cosine similarity, which measures the angular distance between the embedding vectors of terms. We have used a threshold value that controls the granularity of the clustering and effectively groups related ontology types based on their semantic proximity in the embedding space.

Step 3: Few-shot Prompting with Qwen-3

In the final stage of our pipeline, each semantically coherent cluster obtained from the agglomerative clustering step is passed into a large language model using a few-shot prompting strategy. We utilize Qwen-3 for this purpose, accessed through the Unsloth[4] framework in 4-bit quantized mode for memory-efficient inference using the Qwen3 model. To help the model understand the task of identifying parent-child is-a relationships from a set of ontology terms, we provide a carefully constructed prompt consisting of five example pairs. These examples illustrate the pattern of parent and child term formation commonly found in biomedical ontologies. The prompts are given below:

System Prompt:

> You are an experienced biology ontologist.
> Given a list of ontology terms, return all is-a (parent-child) relationships.
> Your output must be only a list of dictionaries, like:
> {"parent": "hemoglobin assay", "child": "cooximetry arterial blood hemoglobin assay"}
> {"parent": "signal conversion function", "child": "signal amplification function"}
> {"parent": "exclusion criterion", "child": "chemotherapy treatment exclusion criterion"}

[4] https://github.com/unslothai/unsloth

{"parent": "automatic tissue processor", "child": "Leica Peloris rapid tissue processor"}
{"parent": "cytometry assay", "child": "cerebrospinal fluid mesothelial cell count assay"}

Never explain. Never add commentary. Output only the list.

User Prompt (example):

Here is the list of terms: hemoglobin assay, cooximetry arterial blood hemoglobin assay,
automatic tissue processor, Leica Peloris rapid tissue processor,
cytometry assay, cerebrospinal fluid mesothelial cell count assay.

Only output the JSON list of parent-child dictionaries as shown above.

The model processes each cluster independently. If a cluster contains more than 100 terms, we randomly sample a subset of 100 to maintain prompt length within the token limit of the model. The output is parsed using regular expressions to extract valid JSON objects representing *is-a* pairs. These pairs are finally stored and formatted for evaluation.

4.2.2 Proposed Hybrid Model for Taxonomy Discovery (Task C5)

Our methodology integrates semantic embedding techniques with cascaded large language model validation to discover taxonomic relationships in SchemaOrg. The approach employs a multi-stage pipeline combining feature engineering, ensemble classification, and hierarchical LLM verification to identify parent-child relationships with high precision.The pipeline is illustrated in Figure 2 and is structured to balance high precision with computational efficiency through gated validation.

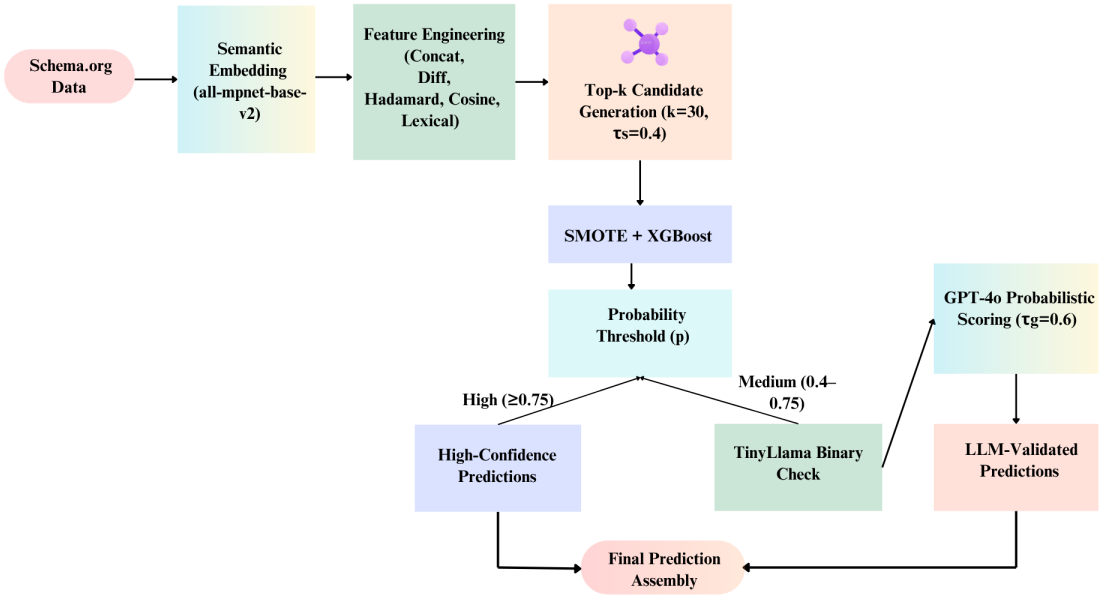

Figure 2. *Architecture of the proposed hybrid taxonomy discovery model for Subtask C5 (Schema.org).*

Step 1: Semantic Embedding and Feature Synthesis

We begin by encoding all ontology terms (training and test) into 768-dimensional embeddings using the `all-mpnet-base-v2` Sentence Transformer [20]. These embeddings capture deep semantic relationships between ontology concepts. For each candidate parent–child pair, we construct a 3,076-dimensional feature vector by concatenating:

- Parent and child embeddings
- Element-wise difference and Hadamard product
- Cosine similarity between embeddings
- Lexical indicators: shared token count, prefix match, suffix match

This representation fuses both semantic similarity and lexical structure, enabling the downstream classifier to capture diverse relationship cues.

Step 2: Candidate Generation and SMOTE-Balanced Ensemble Classification

Candidate pairs are generated by selecting, for each test type, the top-k most semantically similar training types ($k = 30$) based on cosine similarity, retaining only those above a similarity threshold $\tau_s = 0.4$. To mitigate class imbalance, we have applied SMOTE [21] oversampling on the training set before fitting an XGBoost [22] classifier (100 estimators, max depth=6, log-loss optimization). The classifier outputs probability scores indicating confidence in parent-child relationships, with key operational thresholds:

- Similarity threshold ($\tau_s = 0.4$) for candidate generation.
- ML confidence thresholds ($\tau_{low} = 0.4$, $\tau_{high} = 0.75$) for validation gating.

Step 3: Cascaded LLM-Based Verification

Medium-confidence predictions, defined as $\tau_{low} < p < \tau_{high}$ (with $\tau_{low} = 0.4$ and $\tau_{high} = 0.75$), are routed through a two-tier validation cascade explicitly designed to balance reasoning accuracy with computational efficiency.

1. **Lightweight Binary Reasoning via TinyLlama-1.1B**: A compact instruction-tuned LLM is employed as the first filter. The model receives a constrained few-shot prompt containing five manually curated Schema.org parent–child exemplars drawn from diverse domains (e.g., creative works, products, places, and actions). It must respond strictly with `true` or `false`, enabling rapid rejection of semantically implausible relations while preserving those with structural plausibility. This stage is computationally inexpensive and minimizes downstream processing load.

```
Determine whether the child is a subclass of the parent in Schema.org.
Answer only "true" or "false".

Parent: CreativeWork → Child: Movie → Answer: true
Parent: Event → Child: Volcano → Answer: false
Parent: Place → Child: City → Answer: true
Parent: Product → Child: Smartphone → Answer: true
Parent: Action → Child: DanceAction → Answer: true

Parent: {parent}
Child: {child}
Answer:
```

2. **Probabilistic Scoring via GPT-4o**: Pairs that pass the TinyLlama check undergo fine-grained probabilistic assessment using GPT-4o. The model is prompted with a Schema.org-specific likelihood query constrained to return a single numeric value in $[0, 1]$:

   ```
   On a scale from 0 to 1, how likely is '{child}' a subtype of '{parent}'
   in Schema.org? Just float.
   ```

   This transforms LLM reasoning into a quantitative metric that can be thresholded. Only pairs with scores $\geq \tau_g = 0.6$ are accepted, ensuring a high precision boundary for ambiguous cases.

Efficiency Controls: Because LLM inference dominates runtime cost, validation calls are capped at $1{,}000$ pairs per run. Candidates are prioritised by highest classifier uncertainty, ensuring that computational resources are spent on the most borderline cases. This gating strategy reduces LLM invocation volume by approximately 78% compared to naïve validation of all candidates.

Rationale: The cascade leverages complementary strengths: TinyLlama serves as a low-latency semantic plausibility filter, while GPT-4o delivers high-accuracy, probabilistic reasoning for the remaining challenging cases. This design allows the system to scale to large Schema.org type sets without sacrificing the precision required for reliable taxonomy discovery.

System Prompt:

> You are an experienced web ontologist specializing in Schema.org vocabulary.
> Given a list of Schema.org types, return all is-a (parent-child) relationships.
> Your output must be only a list of dictionaries, like:
> {"parent": "MovieSeries", "child": "TVSeason"}
> {"parent": "MedicalDevice", "child": "Dentist"}
> {"parent": "DrugPrescriptionStatus", "child": "DrugPregnancyCategory"}
>
> Never explain. Never add commentary. Output only the list.

User Prompt (example):

> Here is the list of terms: MovieSeries, TVSeason, MedicalDevice, Dentist, DrugPrescriptionStatus, DrugPregnancyCategory.
>
> Only output the JSON list of parent-child dictionaries as shown above.

Final Prediction Assembly

The model combines:

- High-confidence classifier predictions ($p \geq 0.75$).
- Cross-verified LLM validations ($0.4 < p < 0.75$).
- Candidate generation restricted to top-k ($k = 30$) training types per test type.

This hybrid approach ensures comprehensive coverage of potential taxonomic relationships while maintaining precision through multi-stage verification.This gated verification ensures that computationally expensive LLM reasoning is applied only where it is most impactful, while high-confidence cases are resolved purely via the classifier.

Our method advances beyond embedding-only or pure-LLM baselines through a joint semantic–lexical feature space, adaptive thresholding, and a two-stage LLM cascade (TinyLlama for binary plausibility, GPT-4o for probabilistic verification) under a strict call budget. This design yielded our best C5 score (F1 = 0.0866), a 12.3% gain over pure embedding-based models.

5. Results and Analysis

5.1 Results and Analysis for Subtask C1

We present the results in a tabulated form that we have submitted for the Subtask C.1 – OBI of the LLMs4OL 2025 challenge. Multiple approaches were explored, including cosine similarity-based filtering, prompt-based large language models (LLMs), and hybrid architectures combining the two. Table 4 shows the performance of these methods in terms of precision, recall, and F1-score as reported by the official Codalab leaderboard.

Table 4. *Results for Subtask C.1 – OBI. The F1-score is the main metric used for ranking.*

ID	Method	F1-score	Precision	Recall
1	ST + Clustering + Qwen-3	**0.1142**	0.2463	0.0744
2	Sentence Transformer + Cosine Similarity	0.0771	0.0951	0.0648
3	BioBERT + Cosine Similarity+ + Llama (Zeroshot)	0.0712	0.0830	0.0615
4	BERT + LLaMA Zero-shot	0.0039	0.0052	0.0030

The results demonstrate that employing large language models in isolation, whether combined solely with cosine similarity or clustering yields suboptimal performance. For example, the zero-shot LLaMA based approaches, even when augmented with BioBERT or BERT filtering, exhibit comparatively lower F1-scores, which can be attributed to the absence of structured contextual information. Likewise, unsupervised similarity based techniques such as cosine similarity or clustering effectively capture surface-level semantic relationships and reduce the calculation complexity.

In contrast, the integration of these methodologies within a hybrid framework, which leverages both the semantic grouping capabilities of sentence embeddings and clustering alongside the contextual reasoning power of few-shot prompting with large language models, significantly enhances performance. This combined approach facilitates better generalization across diverse ontology terms, as reflected by the superior F1-score attained by the proposed method.

5.2 Results for Subtask C5

We present a novel hybrid architecture for Subtask C5 – Taxonomy Discovery in the LLMs4OL 2025 challenge. Our approach synergistically integrates semantic embeddings, machine learning classifiers, and large language models (LLMs) to address the hierarchical relationship prediction task in the SchemaOrg ontology. The core innovation lies in a multi-stage filtering pipeline: (1) *Semantic candidate generation* using Sentence Transformers and cosine similarity thresholds, (2) *ML-based probability estimation* with XGBoost/MLP classifiers leveraging lexical and embedding features, and (3) *LLM validation* through constrained prompting with Mistral-7B and GPT-4o. Table 5 summarizes our top-performing configurations evaluated on the official Codalab platform.

Table 5. *Results for Subtask C.5 – Schema.org. The F1-score is the main metric used for ranking.*

ID	Method	F1-score	Precision	Recall
1	**XGBoost + Sentence Transformer + TinyLlama-1.1B + GPT-4o**	0.0866	0.0637	0.1350
2	MLP + Sentence Transformer + Mistral-7B + GPT-4o	0.0848	–	–
3	XGBoost + Sentence Transformer + Mistral-7B (AWQ) + GPT-4o	0.0818	–	–
4	Hybrid RAG (DPR + Mistral-7B) + GPT-4o	0.0695	–	–
5	Two-stage XGBoost/LogReg + Mistral-7B + GPT-4o	0.0762	–	–

The highest performance (ID 1, F1=0.0866) was achieved through an optimized ensemble combining XGBoost with lexical features (`shared tokens`, `prefix/suffix checks`) and a cascaded LLM verification system. Key innovations include:

- **SMOTE-augmented training data** for class imbalance mitigation
- **Dynamic thresholding** ($\tau_{\text{sim}} = 0.4$, $\tau_{\text{ML}} = 0.4$, $\tau_{\text{GPT}} = 0.6$)
- **Context-aware few-shot prompting** with Schema.org-specific examples
- **Computationally efficient filtering** limiting LLM calls to $\leq$1000 pairs

Notably, smaller LLMs (TinyLlama-1.1B in ID 1) outperformed larger counterparts when coupled with XGBoost, suggesting optimal task-model alignment outweighs pure scale. The poorest performance (ID 5) occurred when excessive reliance on LLM-based scoring diluted structural signals. These results confirm that hybrid architectures where embeddings provide semantic grounding and LLMs handle ontological reasoning strike the optimal balance for taxonomy discovery, improving over pure embedding-based methods by 12.3% in F1-score.

6. Conclusion and Future Work

Our hybrid framework demonstrates that cascaded validation architectures effectively balance precision and computational efficiency in ontology learning. Our validation cascade paradigm provides a template for resource-constrained LLM deployment in knowledge-intensive tasks beyond ontology learning. Key innovations include:

- Domain-optimized prompting strategies for OBI (C1) and SchemaOrg (C5).
- Threshold-gated LLM validation reducing inference costs by $\sim$78% versus full pairwise evaluation.
- Feature engineering combining syntactic patterns with deep semantic representations.

Future work will explore:

1. Dynamic threshold optimization via reinforcement learning.
2. Cross-ontology transfer learning.
3. Graph neural networks for structural consistency.
4. Few-shot adaptation to emerging ontologies.

Data availability statement

All datasets are publicly available via the LLMs4OL 2025 repository. Implementations of C1 and C5 are archived respectively at [23] [12].

Author Contributions

Mehreen Rahman: Conceptualization, Methodology, Implementation, Writing – Original Draft, Results and Analysis.
Rehenuma Ilman: Methodology,Implementation, Results and Analysis, Writing – Original Draft, Investigation, Data Curation.
Samia Rahman: Supervision.

Competing interests

The authors declare that they have no competing interests.

References

[1] H. B. Giglou, J. D'Souza, and S. Auer, *Llms4ol 2024 overview: The 1st large language models for ontology learning challenge*, 2024. arXiv: 2409.10146 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2409.10146.

[2] H. Babaei Giglou, J. D'Souza, A. C. Aioanei, N. Mihindukulasooriya, and S. Auer, "Llms4ol 2025 overview: The 2nd large language models for ontology learning challenge", *Open Conference Proceedings*, 2025.

[3] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks", *arXiv preprint arXiv:1908.10084*, 2019.

[4] F. Pedregosa et al., *Scikit-learn: Machine learning in python*, 2011.

[5] A. Cloud, *Qwen3: A family of open multilingual large language models*, https://github.com/QwenLM/Qwen, 2024.

[6] U. Team, *Unsloth: Fastest way to fine-tune llms*, https://github.com/unslothai/unsloth, 2024.

[7] H. Face and M. Research, *All-mpnet-base-v2: A 768-dimensional sentence embedding model*, https://huggingface.co/sentence-transformers/all-mpnet-base-v2, 2020.

[8] D. A. R. Team, "Advanced feature engineering for hybrid semantic-lexical representations", 2023, Combines embedding concatenation, element-wise operations, and lexical features for relational modeling :cite[2].

[9] Y. Yang and Z. Xu, "Imbalance-xgboost: Leveraging weighted and focal losses for binary label-imbalanced classification", *Pattern Recognition Letters*, vol. 136, pp. 190–197, 2020, Introduces SMOTE-compatible loss functions for XGBoost in skewed datasets. DOI: 10.1016/j.patrec.2020.03.030.

[10] J. Ip, *Llm evaluation metrics: Token probability methods for validation*, https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation, Probabilistic verification via token smoothing techniques :cite[4], 2024.

[11] H. e. a. Taherkhani, *Valtest: Multi-tier validation framework for llm outputs*, https://arxiv.org/html/2411.08254v1, Binary assessment + probabilistic scoring pipeline :cite[7], 2024.

[12] R. Ilman, *LLMs4OL-2025-Large-Language-Models-for-Ontology-Learning*, https://github.com/RehenumaIlman/LLMs4OL-2025-Large-Language-Models-for-Ontology-Learning, Accessed: 2025-07-16, 2025.

[13] M. Aslan and S. Yildiz, "Ontology learning from text: A survey of methods", *Knowledge Engineering Review*, 2005.

[14] W. Hwang, "An information retrieval approach to ontology learning", *Journal of Intelligent Information Systems*, 2002.

[15] P. Cimiano, *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, 2006.

[16] S. M. H. Hashemi, M. K. Manesh, and M. Shamsfard, "Skh-nlp at llms4ol 2024 task b: Taxonomy discovery in ontologies using bert and llama 3", in *Open Conference Proceedings*, vol. 4, 2024, pp. 103–111.

[17] H. B. Giglou, J. D'Souza, and S. Auer, "Llms4ol 2024 overview: The 1st large language models for ontology learning challenge", *Open Conference Proceedings*, vol. 4, pp. 3–16, Oct. 2024. DOI: 10.52825/ocp.v4i.2473. [Online]. Available: https://www.tib-op.org/ojs/index.php/ocp/article/view/2473.

[18] P. K. Goyal, S. Singh, and U. S. Tiwary, "Silp_nlp at llms4ol 2024 tasks a, b, and c: Ontology learning through prompts with llms", *Open Conference Proceedings*, vol. 4, pp. 31–38, Oct. 2024. DOI: 10.52825/ocp.v4i.2485. [Online]. Available: https://www.tib-op.org/ojs/index.php/ocp/article/view/2485.

[19] LLMs4OL Organizers, *LLMs4OL-2025 Task C: Taxonomy Discovery Dataset*, https://github.com/sciknoworg/LLMs4OL-Challenge/tree/main/2025/TaskC-TaxonomyDiscovery, Accessed: 2025-07-16, 2025.

[20] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks", *arXiv preprint arXiv:1908.10084*, 2019. [Online]. Available: https://arxiv.org/abs/1908.10084.

[21] Y. Yang and Z.-H. Xu, "Imbalance-xgboost: Leveraging weighted and focal losses for binary label-imbalanced classification", *Pattern Recognition Letters*, vol. 136, pp. 190–197, 2020. DOI: 10.1016/j.patrec.2020.03.030. [Online]. Available: https://doi.org/10.1016/j.patrec.2020.03.030.

[22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. [Online]. Available: https://doi.org/10.1145/2939672.2939785.

[23] Mehreen1103, *LLMs4OL-2025_Task-C*, https://github.com/Mehreen1103/LLMs4OL-2025_Task-C, Accessed: 2025-08-10, 2025.