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.2901

Published: 01 Oct. 2025

SEMA at LLMs4OL 2025 Task C: Prompt-Decoupled Fine-Tuning on MatOnto with LLaMA

Miquel Canal-Esteve[1,*], José Abreu-Salas[1], and Yoan Gutiérrez[1]

[1]University of Alicante, Spain

*Correspondence: Miquel Canal-Esteve, mikel.canal@ua.es

Abstract. This paper presents our submission to Task C (Relation Extraction) of the LLMs4OL 2025 Challenge, which investigates the ability of Large Language Models (LLMs) to identify semantic and taxonomic relations between ontology types. Focusing on the MatOnto subtask—selected for its manageable size—we explore the performance of open-source models under resource constraints. We fine-tune LLaMA 3.1–8B using LoRA adapters and evaluate various strategies including contrastive negative sampling, prompt inversion, and system prompt variation. Inspired by recent findings on prompt sensitivity, we adopt a cross-template setup where the model is trained with one prompt format and tested with another semantically equivalent variant. Our experiments suggest that prompt-decoupling can improve generalization and mitigate overfitting to specific phrasings. While our results are modest, they offer insights into the challenges of adapting LLMs to structured relation extraction tasks and highlight practical considerations for tuning under constrained resources.

Keywords: Ontology Relation Extraction, Prompt Generalization, LoRA Fine-Tuning

1. Introduction

This paper presents our contribution to the LLMs4OL 2025 Challenge, an extension of the paradigm introduced by Babaei et al. (2023) [1], which investigates how Large Language Models (LLMs) can support Ontology Learning (OL) tasks such as term typing, taxonomy discovery, and relation extraction. While the original paper formalizes these three core tasks, the 2025 challenge introduces a fourth task (Task D) focused on ontology verbalization, expanding the scope of the framework. Our work addresses Task C: Relation Extraction, which consists of determining whether a semantic or taxonomic relationship holds between two given types.

The LLMs4OL 2025 Challenge defines eight subtasks within Task C, each corresponding to a different knowledge domain—such as geography, biomedicine, and materials science. Due to our limited computational resources (3× A100 GPUs with 40 GB each in a shared environment with the research group), we selected the MatOnto subtask, which is the smallest among the eight and thus more suitable for experimentation with open-source models.

We fine-tune LLaMA 3.1–8B using LoRA [2] adapters to reduce training costs. We also experimented with LLaMA 3.2–3B and LLaMA 3.1–8B Instruct, but the base

8B model yielded consistently superior results. Although we explored a variety of strategies—including data augmentation with negative samples, WordNet integration, and prompt variation—we focus in this document on the configurations that delivered the most effective performance. All code, data splits, and evaluation scripts are available in our public repository at https://github.com/miquelcanalesteve/LLMs4OL-SEMA.

2. Related Work

Ontology Learning (OL) aims to automate the extraction of ontological structures—such as types, taxonomies, and semantic relations—from unstructured or semi-structured data. Traditional approaches often relied on lexico-syntactic pattern mining and rule-based systems, but recent advances in Large Language Models (LLMs) have introduced more flexible, prompt-driven alternatives[1].

In the context of Task C: Relation Extraction, two recent participant papers from the LLMs4OL 2024 Challenge provide valuable insights. The RWTH-DBIS team [3] applied domain-specific continual pretraining and task-specific fine-tuning on open-source models like LLaMA, incorporating context-enriched prompts built from external sources such as Wikipedia. Their results showed that injecting structured context during training could enhance the model's ability to infer type relations.

The SKH-NLP team [4] proposed a method based on prompt engineering and data augmentation with negative examples, particularly for taxonomic relation prediction in the GeoNames dataset. They systematically generated negative samples by reversing parent-child pairs or randomly replacing one of the terms with an unrelated type, which improved binary classification performance. They also experimented with a variety of prompt templates, demonstrating that even minor surface-level changes in wording (e.g., "parent class" vs. "superclass") could cause large behavioural shifts in model predictions.

Beyond these empirical insights from prior challenge participants, recent research has also highlighted the value of decoupling prompt formulations used during training and inference. For instance, the PTST principle proposed by Lyu et al. (2024) [5] suggests that training without alignment-specific prompts and introducing them only at test time can preserve generalization. Similarly, Sclar et al. (2023) [6] show that LLMs are highly sensitive to minor changes in prompt formatting, even when semantics are preserved—highlighting the risk of overfitting to superficial features.

Motivated by these findings, our work adopts a strategy where the model is fine-tuned using one prompt formulation and evaluated using another that expresses the same underlying relation in an inverted form. This variation tests the model's ability to generalize beyond surface prompt features. Combined with LoRA-based fine-tuning, contrastive negative sampling, and experimentation under constrained resources, this forms the core of our methodology for the MatOnto subtask in Task C.

3. Methodology

Our approach to Task C: Relation Extraction in the LLMs4OL 2025 Challenge is designed to be efficient and adaptable, leveraging lightweight fine-tuning, prompt variation, and data augmentation strategies. The main components of our methodology are described below. To provide a concise overview, Algorithm **??** summarizes the entire pipeline.

3.1 Model and Fine-tuning Setup

We use the LLaMA 3.1–8B base model as our backbone, applying parameter-efficient fine-tuning with LoRA adapters [2]. LoRA introduces low-rank adaptation matrices into selected attention and MLP modules, allowing the model to be trained efficiently by updating only a small subset of parameters. We set the LoRA rank $r = 8$, scaling factor $\alpha = 16$, and dropout rate to 0.05. The target modules include the self-attention projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and the feed-forward network projections (`gate_proj`, `up_proj`, `down_proj`).

Fine-tuning was performed on 3$\times$ A100 GPUs (40 GB each, shared environment with the research group) using the Fully Sharded Data Parallel (FSDP) strategy provided by Lightning Fabric, which allows memory-efficient distribution of the model across GPUs. The optimizer used is AdamW with a learning rate of 2×10^{-5} and a micro-batch size of 2 per GPU. Training was capped at 30 epochs with early stopping (patience of 3 epochs) based on validation loss. A fixed random seed (42) ensured reproducibility.

In addition to the base model, we also ran comparative experiments using LLaMA 3.1–8B Instruct to assess whether models with built-in alignment perform better out-of-the-box. However, in our setting, this model showed lower performance.

Tokenization.

Before training, all datasets are preprocessed and tokenized using the official LLaMA tokenizer. Instruction–response examples are formatted using a configurable prompt template system that constructs input sequences in the form:

```
System: <system_message>
User: <instruction>
Assistant: <output>
```

The prompt and response are concatenated and tokenized with a maximum length of 512 tokens. The model is trained to predict only the response by masking out the prompt tokens in the label sequence using an ignore index of -100. Padding is applied to the right based on the longest sequence in each batch, and sequences exceeding the maximum length are truncated. Tokenized datasets are saved using the Hugging Face `datasets` format to ensure consistent preprocessing and efficient loading during training.

3.2 Negative Sampling and Data Augmentation

The training dataset consists of a JSON file defining explicit `parent-child` semantic relations between pairs of types. Additionally, a separate list of all 654 types is provided. In theory, each type could be related to every other type (excluding self-relations), leading to:

$$N = 654 \times (654 - 1) = 426{,}282$$

possible ordered type pairs. However, the dataset contains only 840 annotated positive relations, resulting in a high sparsity and an approximate ratio of:

$$\text{Positive ratio} = \frac{840}{426{,}282} \approx 1 : 507$$

This extreme imbalance motivates the need for contrastive training with negative examples. Following the approach of SKH-NLP [4], we introduced challenging false examples by randomly replacing parents or children while ensuring no overlap with the gold positives. We tested ratios of 5:1, 6:1, and 7:1 to evaluate how the balance affects model performance.

3.3 Prompt Strategy and Cross-Template Generalization

A key design choice is the use of cross-prompt generalization, where the prompt format used during fine-tuning differs from the one used during inference. While training used the template *"Is 'X' a subclass of 'Y'?"*, inference employed variants such as *"Is 'X' a parent class of 'Y'?"*. This decoupling prevents overfitting to specific wordings and encourages the model to focus on relational semantics rather than surface forms.

3.4 System Prompt Variation

A key design choice in our methodology is the use of cross-prompt generalization, where the prompt format used during fine-tuning differs from the one used during inference. This approach is inspired by the broader principle of decoupling training and testing prompts introduced in the PTST (Pure Tuning, Safe Testing) framework by Lyu et al. (2024) [5].

Although their work is primarily motivated by safety alignment—advocating for omitting safety prompts during fine-tuning and applying them only at inference time—the underlying insight transfers to our setting: relying too heavily on a fixed prompt structure during training may cause the model to specialize to surface-level features, thereby reducing generalization at test time. Instead, we adopt a prompting strategy that preserves the underlying relational logic while deliberately varying the surface form and directionality between training and inference.

In parallel, Sclar et al. (2023) [6] demonstrate that LLMs are highly sensitive to superficial prompt formatting changes, including punctuation, phrasing, and token ordering. Even when semantic content is preserved, such variations can yield drastically different outputs—suggesting that surface-level prompt specialization can lead to brittle models. These findings further motivate our strategy of deliberately varying the prompt template between training and inference, while preserving the underlying task semantics.

We experimented with the following templates:

- *"Is 'X' a subclass of 'Y'? Answer with 'true' or 'false'."*
- *"Is 'X' a parent class of 'Y'? Answer with 'true' or 'false'."*
- *"Is 'X' a superclass of 'Y'? Answer with 'true' or 'false'."*

3.5 Overall Workflow in Pseudocode

The entire methodology, from dataset preparation to evaluation, is summarized below.

4. Experiments and Evaluation

Numerous experiments were conducted using the training dataset to identify the most effective configuration; however, we only report the results submitted to Codalab on the official test set.

Algorithm 1 Overall Workflow for Task C: Relation Extraction (MatOnto)

Initialize: LLaMA 3.1–8B base + LoRA (r=8, α=16, dropout=0.05), FSDP training on 3×A100, AdamW (2×10^{-5}), max 30 epochs, early stopping=3.
Step 1: Tokenization
Format as `System: <msg>, User: <inst>, Assistant: <out>`
Tokenize
Step 2: Negative Sampling
Generate negatives by replacing parent/child, ratios {5:1, 6:1, 7:1}.
Step 3: Prompt Strategy
Train with "subclass" prompt, infer with "parent" or "superclass" prompts.
Step 4: System Prompts
Compare *Generic expert* vs. *Material science expert* system roles.
Step 5: Fine-tuning
for each dataset ratio and system prompt **do**
    Fine-tune model with LoRA
    Save checkpoint $M_{r,sp}$
end for
Step 6: Inference
for each saved model and test pair (X, Y) **do**
    Build inference prompt using a different template
    Predict $\in$ {true,false}
end for
Step 7: Evaluation
Compute Precision, Recall, F1; select best configuration

Data Augmentation Ratio

We tested three different false-to-true ratios: 5:1, 6:1, and 7:1. The prompt used during training was "Is X a subclass of Y?", while inference was performed with "Is X a parent class of Y?". The system prompt specified an expert in material science.

Table 1. *LLaMA 3.1–8B with different negative sampling ratios (Material Science system prompt)*

Ratio	F1	Precision	Recall
5 false - 1 true	0.132	0.091	0.241
6 false - 1 true	**0.144**	**0.109**	0.213
7 false - 1 true	0.115	0.080	0.205

The best performance was obtained with the 6:1 ratio, which we used as the default configuration in subsequent experiments.

System Prompt Comparison

We initially developed and tuned our method using a domain-specific system prompt tailored to material science. To assess whether a more generic instruction could also perform well, we compared it against a general-purpose expert prompt. This evaluation was conducted using a fixed false-to-true ratio of 6:1, which had previously shown the best performance among the ratios tested (see Section **??**). As shown in Table 2, the generic prompt achieved higher recall, but the material science prompt yielded better precision and the highest F1 score overall.

Since the domain-specific prompt consistently outperformed the generic one under this optimal ratio, we did not extend the comparison to other ratios. Our goal was to identify the best configuration for final submission, and further exploration across system prompts and ratios was deprioritized in favor of consolidating results on the most promising setup.

Table 2. *LLaMA 3.1–8B with different system prompts (6:1 ratio)*

System prompt style	F1	Precision	Recall
Generic	0.136	0.087	**0.310**
Material science	**0.144**	**0.109**	0.213

Prompt Configuration Between Training and Inference

We investigated how the combination of prompts used in training and inference affects performance. All configurations used the material science system prompt and a 6:1 negative-to-positive ratio.

Table 3. *Prompt configuration effects: train → inference. The row Subclass → Parent (Instruct) corresponds to the LLaMA 3.1–8B Instruct model.*

Train → Inference	F1	Precision	Recall
Subclass → Subclass	0.043	0.027	0.116
Subclass → Parent	**0.144**	**0.109**	0.213
Subclass → Parent (Instruct)	0.135	0.086	**0.310**
Parent → Parent	0.011	0.018	0.008
Parent → Subclass	0.007	0.005	0.011
Subclass → Superclass	0.111	0.070	0.271

These results confirm that prompt decoupling—using different but semantically equivalent prompt formats during training and inference—substantially improves performance. The *subclass → parent* configuration consistently outperformed others, while using the same prompt at both stages led to overfitting and poorer generalization.

We initially selected the LLaMA 3.1–8B **base** model for fine-tuning because early exploratory experiments using the training set showed more stable and interpretable learning behavior compared to the Instruct variant. Additionally, fine-tuning the base model allowed us to exert greater control over the formatting and semantics of the prompt–response structure.

To validate the robustness of our best configuration, we tested it using the LLaMA 3.1–8B **Instruct** model without further tuning. Despite its zero-shot capabilities, the Instruct model underperformed slightly in F1 compared to the fine-tuned base model, although it achieved the highest recall. Due to time and computational constraints (3× A100 GPUs in a shared environment), we limited our experimentation with Instruct to this evaluation. Future work could include fine-tuning Instruct or hybrid strategies combining both model types.

5. Discussion

Our experiments reveal several key insights into training LLMs for ontology relation extraction under resource-constrained conditions:

Prompt Decoupling Improves Generalization

The most striking result is the underperformance of identical train–inference prompt pairs. Models fine-tuned and evaluated on the same template (e.g., *subclass* $\rightarrow$ *subclass* or *parent* $\rightarrow$ *parent*) performed significantly worse than cross-template setups. Models trained on *subclass* and evaluated on *parent* achieved the best performance, suggesting that decoupling the surface form of prompts fosters generalization. This aligns with the PTST principle [5] and supports our hypothesis that prompt specialization can lead to overfitting on syntactic patterns rather than semantic understanding.

System Prompt Framing Has Marginal but Consistent Effects

Although differences were moderate, domain-specific system prompts (e.g., "expert in material science") consistently outperformed generic prompts. This suggests that expert framing can guide the model toward more domain-sensitive interpretations, even when no additional knowledge is injected.

Negative Sampling Ratio Needs Careful Tuning

Our results indicate that the number of negative examples plays a crucial role in model performance. A 6:1 false-to-true ratio offered the best balance between training signal and label imbalance. Higher ratios (e.g., 7:1) reduced performance, likely due to noise overwhelming the limited set of true examples. These findings emphasize the importance of calibrating contrastive data augmentation.

Prompt Semantics Matter More Than Syntax

Prompt variants that violated the underlying semantics of the task (e.g., *parent* $\rightarrow$ *subclass*, or *subclass* $\rightarrow$ *synonym*) resulted in sharp performance drops. This confirms that cross-template generalization only works when semantic alignment is preserved, even if surface phrasing differs. It reinforces the idea that models must internalize relational logic rather than rely on superficial prompt cues.

Instruction-Tuned Variants Show High Recall

Interestingly, the LLaMA 3.1–8B Instruct model, despite underperforming during training, achieved strong recall when used during inference with prompt decoupling. This suggests that instruction-tuned models may have a helpful inductive bias toward answering questions reliably, though further work is needed to confirm this behavior under different prompting and tuning configurations.

6. Limitations

This study is subject to several limitations. First, our experiments were conducted under strict computational constraints (3$\times$ A100 GPUs in a shared environment), which limited the number of models, prompt variations, and hyperparameter combinations we could explore. Second, the focus on the MatOnto subtask—chosen for its smaller size—may restrict the generalizability of our findings to other domains in Task C with different relational patterns and term distributions. Third, although we experimented with multiple prompt-decoupling configurations, we did not perform a full ablation across all possible semantic-preserving variations, leaving open questions about the optimal degree of

prompt diversity. Fourth, our evaluation relied solely on the official challenge metrics and test sets, without additional error analysis or human validation to assess the qualitative nature of model predictions. Finally, the extreme class imbalance in the dataset (1:507 ratio of positives to negatives) may have amplified the sensitivity of results to specific negative sampling ratios, and alternative sampling or loss-balancing strategies remain unexplored.

7. Conclusion and Future Work

In this work, we presented our approach to Task C (Relation Extraction) of the LLMs4OL 2025 Challenge, focusing on the MatOnto subtask. We explored how open-source LLMs—specifically LLaMA 3.1–8B fine-tuned with LoRA adapters—can be adapted for ontology relation extraction under limited computational resources. Our methodology combined prompt decoupling, contrastive negative sampling, and system prompt variation to improve generalization and robustness.

Our experiments highlight the effectiveness of training and inference prompt decoupling. The best performance was obtained when prompts were semantically aligned but structurally distinct across training and test phases. In contrast, identical prompts at both stages led to clear overfitting and performance degradation. Additionally, moderate contrastive sampling (6:1 ratio) proved beneficial, while excessive negative examples harmed learning. Small but consistent improvements were also observed when using domain-specific expert prompts.

For future work, we plan to extend this study in several directions. First, we aim to continue investigating prompt decoupling by exploring additional prompt variations, evaluating robustness across domains, and analysing model sensitivity to surface-level changes. Second, we intend to incorporate structured semantic data during training—potentially through continual pretraining—while paying special attention to data quality and domain alignment. Finally, we are interested in hybrid approaches that combine LLM-driven extraction with human validation and downstream inference. In particular, we envision a retrieval-augmented generation (RAG) pipeline where the model proposes candidate relations, retrieves relevant context, and refines its output with reasoning support from curated knowledge.

Data availability statement

The data used in this study corresponds to the official datasets provided by the organizers of the LLMs4OL 2025 Challenge. All datasets are publicly available at https://github.com/sciknoworg/LLMs4OL-Challenge/tree/main/2025. No additional third-party or proprietary data was used.

Underlying and related material

All code used to fine-tune the models, generate negative samples, and run the experiments is publicly available in our GitHub repository: https://github.com/miquelcanalesteve/LLMs4OL-SEMA. The repository includes training scripts, evaluation metrics, and configuration files to reproduce the results presented in this paper.

Author contributions

Miquel Canal-Esteve: Conceptualization, Methodology, Software, Investigation, Data Curation, Formal Analysis, Writing – Original Draft, Writing – Review & Editing. José Abreu-Salas: Methodology, Investigation, Formal Analysis, Writing – Review & Editing. Yoan Gutiérrez: Supervision, Methodology, Writing – Review & Editing.

All contributions are reported in accordance with the CRediT (Contributor Roles Taxonomy) guidelines.

Competing interests

The author declares that they have no competing interests.

Funding

This work did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgements

The author thanks the organizers of the LLMs4OL 2025 Challenge for providing the datasets and evaluation framework that made this research possible.

References

[1] 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.

[2] E. J. Hu et al., "Lora: Low-rank adaptation of large language models.", *ICLR*, vol. 1, no. 2, p. 3, 2022.

[3] 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", in *Open Conference Proceedings*, vol. 4, 2024, pp. 49–63.

[4] 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.

[5] K. Lyu, H. Zhao, X. Gu, D. Yu, A. Goyal, and S. Arora, "Keeping llms aligned after fine-tuning: The crucial role of prompt templates", *arXiv preprint arXiv:2402.18540*, 2024.

[6] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr, "Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting", *arXiv preprint arXiv:2310.11324*, 2023.