LLMs4OL 2026: The 3rd Large Language Models for Ontology Learning Challenge at the 25th ISWC
LLMs4OL 2026 Task Participant Papers
TIB-OP will set DOI with \TIBdoi

Submitted: 2026-07-26

LettRAGraph at LLMs4OL 2026 Task Flagship: Prompt-Guided Ontology Extraction with Ensembling

Pasquale Lisena[1], Julien Plu[2], Oscar Moreno Escobar[2], Edouard Trouillez[2], and Raphael Troncy[1]

[1]EURECOM, Sophia Antipolis, France
[2]LettrIA, Paris, France
*Correspondence: Pasquale Lisena, pasquale.lisena@eurecom.fr ; Julien Plu, julien@lettria.com

Abstract: We present a system for constructing ontology graphs from text using open-weight large language models. Our approach combines structured prompting, predicate normalization, and structural filtering. Prompt ablation experiments show that explicit taxonomy rules and closed-world constraints improve graph quality. We also introduce a conservative ensemble that retains only triples shared by two prompt configurations. On the official test set, the selected ensemble achieves Graph Similarity scores of 0.3584, 0.4872, and 0.5162 under exact, fuzzy, and semantic matching.

Keywords: ontology learning, knowledge graph extraction, large language models, prompt engineering, ensemble methods

1 Introduction

Ontology engineering remains one of the main bottlenecks in the development and maintenance of knowledge graphs and semantic applications. Constructing high-quality ontologies requires significant domain expertise and manual effort to identify concepts, define their semantics, and organize them into coherent taxonomic structures. Ontology Learning (OL) aims to alleviate this burden by automatically extracting ontology elements from unstructured or semi-structured sources, supporting ontology construction and enrichment [1], [2]. Recent advances in Large Language Models (LLMs) have significantly expanded the capabilities of ontology learning systems, enabling the generation and refinement of ontology components through their broad linguistic and semantic knowledge [3], [4], [5], [6].

The LLMs4OL Challenge addresses this need by providing a common benchmark for assessing the capabilities of LLMs across a range of ontology learning tasks [7]. Participants are evaluated using common datasets, standardized metrics, and reproducible experimental settings, fostering the development of robust and generalizable approaches for ontology learning. The 2026 edition [8] introduces a *Flagship Task* that focuses on the automatic extraction of primitive ontology relations from textual descriptions, with a strong emphasis on taxonomic structure. Given a collection of concepts and their natural language descriptions, participating systems are required to infer hierarchical `subClassOf` relationships and reconstruct the underlying taxonomy. This task

evaluates a system's ability to combine semantic understanding, reasoning, and structured knowledge acquisition. Unlike previous tasks that focused on individual ontology learning components, the Flagship Task provides a more holistic evaluation of LLM-based approaches by assessing their capacity to generate coherent and semantically consistent taxonomic structures from textual evidence.

Our strategy relies on prompting open-weight LLMs to extract knowledge graphs in a structured intermediate representation, followed by a lightweight post-processing stage that normalizes predicates and converts the generated graph into the challenge format. We iteratively refine the extraction prompt to reduce unsupported relations and improve the distinction between taxonomic and instance-level assertions. Since structural predicates account for most benchmark annotations, we also evaluate a structure-only configuration that retains only these relations. Finally, we investigate a conservative ensemble that keeps triples produced by multiple complementary configurations.

Rather than focusing exclusively on maximizing challenge performance, our primary goal is to assess the robustness of our solution and its potential to generalize beyond the specific setting of the benchmark. Our results show that explicit taxonomy-oriented prompting and structural post-processing substantially improve graph quality, while a conservative ensemble achieves the best overall Graph Similarity by reducing spurious relations at the cost of some coverage.

The remainder of this paper is organized as follows. Section 2 introduces the challenge dataset, while our methodology is detailed in Section 3. Section 4 reports the configuration of the experiments, whose results are discussed in Section 5. Finally, we conclude and detail some future work in Section 6.

2 Dataset

The experiments were conducted using the official dataset provided for the Flagship Task of the LLMs4OL Challenge 2026[1]. Each dataset instance consists of a unique identifier, a textual context, and a set of reference ontology triples. The textual context is a synthetic document comprising a title and one or more paragraphs describing concepts from a particular domain. The corresponding output is a set of *primitive ontology triples*, where each triple is represented as $(subject,\ predicate,\ object)$.

The dataset spans a broad range of topical domains, including healthcare, biology, chemistry, geography, history, automotive engineering, business, law, computer science, education, sports, and culture, thereby providing a heterogeneous benchmark for open-domain ontology and knowledge graph extraction.

The benchmark contains 4,303 training instances and 2,018 test instances. The reference triples are provided only for the training set, while the annotations for the test set are withheld by the challenge organizers and used for the final evaluation.

Although the benchmark supports a broad range of ontological relations, the distribution of predicates is highly imbalanced. The `is-a` relation accounts for 89.08% of all triples, while `instance-of` represents an additional 8.77%. Together, these two predicates constitute approximately 98% of all annotations. The remaining relations, including `is defined by`, `equivalent class`, `disjoint with`, `part_of`, and several others, each represent less than 1% of the dataset, as is shown in Table 1.

[1] https://github.com/sciknoworg/LLMs4OL-Challenge/tree/main/2026/TaskA-Flagship

Table 1. *Distribution of the most frequent predicates in the training set.*

Predicate	Count	Percentage
`is-a`	49,532	89.08%
`instance-of`	4,876	8.77%
`is defined by`	351	0.63%
`equivalent class`	284	0.51%
`disjoint with`	261	0.47%
`type`	114	0.21%
Other predicates	175	0.33%
Total	55,593	100.00%

This extreme imbalance indicates that the benchmark is primarily centered on taxonomic knowledge extraction. Consequently, correctly identifying subclass and instance relationships is the dominant factor in the overall evaluation, whereas the contribution of the remaining predicates is comparatively limited. This observation motivates approaches that prioritize accurate extraction of the most frequent relations while carefully considering whether modeling the long tail of infrequent predicates justifies the additional complexity.

3 Methodology

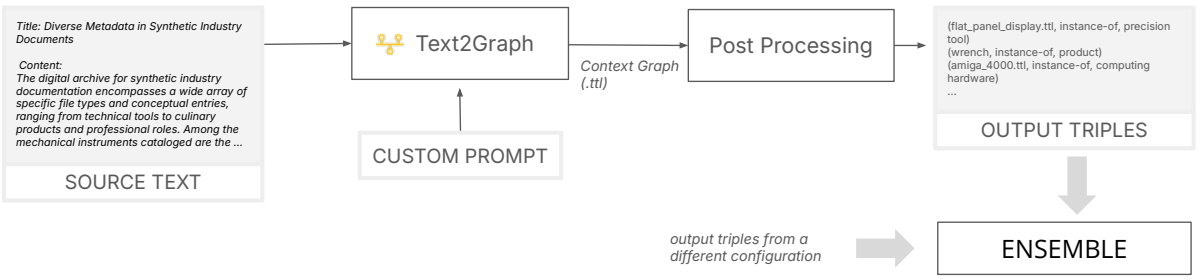

Figure 1. *Overview of the proposed methodology*

Figure 1 illustrates the overall methodology.

We employed **Text2Graph**[2] [9], a prompt-based tool for extracting RDF entities and properties from textual resources using an input prompt (see Section 4.1). The library serializes the extracted RDF graphs in Turtle.

The output produced by Text2Graph is subsequently converted into the triple format required by the challenge. The **post-processing** pipeline consists of the following steps:

- parsing the generated graph into a set of triples;
- converting URI-based identifiers into their corresponding local labels, e.g., `example.org/Person` is reduced to `Person`;
- normalizing `rdfs:subClassOf` and semantically equivalent predicates, such as `example.org/isSubclassOf`, to the challenge predicate `is-a`;
- normalizing `rdf:type` and semantically equivalent predicates, such as `example.org/type`, to `instance-of`;

[2] https://www.lettria.com/features/text-to-graph

- removing self-referential or near-self-referential triples whose subject and object labels have a Levenshtein distance of at most one. This step eliminates outputs such as `(Mineral, is-a, Mineral)` or `(Mineral, partOf, Minerals)`;
- optionally retaining only the two structural predicates, `is-a` and `instance-of`, which together account for approximately 98% of the relations in the dataset.

This optional filtering step addresses a mismatch between the behavior of Text2Graph and the characteristics of the benchmark. Text2Graph tends to generate a comparatively rich set of predicates in order to provide a more complete representation of the input text. However, because relations other than `is-a` and `instance-of` are only sparsely represented in the dataset, generating such predicates may negatively affect the challenge evaluation metrics, as further discussed in the experimental results.

Importantly, the extraction method itself is not restricted to taxonomic relations: in the complete-graph configuration, Text2Graph may generate arbitrary predicates supported by the source text. The restriction to `is-a` and `instance-of` is therefore an evaluation-oriented post-processing choice, whose impact is assessed explicitly by comparing complete-graph and structure-only configurations in Section 5.

The triples produced at this stage already constitute a valid challenge submission. We nevertheless investigated whether combining the outputs of two different configurations could further improve performance. Given two configurations, C_1 and C_2, the **ensemble module** retains only the triples generated by both systems, corresponding to the intersection $T_{C_1} \cap T_{C_2}$ of their respective output sets. This conservative strategy is intended to increase precision by discarding relations supported by only one configuration.

4 Experimental Configurations

4.1 Prompt Engineering

We adopted an iterative prompt engineering strategy to improve the precision of knowledge graph extraction. The initial prompt (A) was inspired by previous work on ontology learning with large language models [10] and progressively refined by introducing additional constraints aimed at reducing unsupported relations and enforcing a consistent representation of taxonomic knowledge.

All prompt variants share the same core structure, consisting of:

- a task definition specifying knowledge graph extraction from text;
- an output constraint requiring a valid JSON object;
- conventions for entity and predicate naming;
- instructions on namespace prefixes;
- a fallback output when no graph can be extracted.

Starting from this common structure, each prompt introduces additional constraints intended to improve extraction quality. Table 2 summarizes the evolution of the prompt variants evaluated in this work.

4.2 Models and Parameters

All experiments were conducted using open-weight large language models, namely Gemma 4 [11] and Qwen 3.6 [12], selected to evaluate both prompt engineering strategies and model capabilities. Unless otherwise stated, all experiments used the default

Table 2. *Summary of the prompt engineering iterations evaluated in this work. Each prompt extends the previous one by introducing additional constraints or clarifications.*

Prompt	Main modification
A	Defines the extraction task, output schema, naming conventions, and fallback behaviour.
B	Explicitly restricts the output to facts directly supported by the input text.
C	Introduces an explicit distinction between `rdf:type` (instances) and `isSubclassOf` (classes), while discouraging synonymous predicates.
D	Changes the preference policy to favour subclass relations when the distinction is ambiguous.
E	Adds strict closed-world constraints, forbidding unsupported facts, intermediate taxonomy nodes, and inferred knowledge, together with an explicit validation step before producing the output.
F	Clarifies the semantics of `rdf:type` and `isSubclassOf` using positive examples and more detailed instructions.
G	Introduces the `rdfs` namespace and modifies the preference policy to favour `rdf:type` in taxonomy-related cases.
H	Replaces the custom taxonomy predicate with the standard `rdfs:subClassOf`.
I	Simplifies the task by restricting the output vocabulary to only two predicates: `rdf:type` and `rdfs:subClassOf`.
J	Reformulates the prompt as a precision-first closed-world extraction task with detailed validation rules to minimize hallucinated relations.
K	Further strengthens Prompt J by explicitly defining acceptable evidence for taxonomy and instance assertions, including structural taxonomy cues while reinforcing the closed-world assumptions.

decoding configuration with a temperature of $T = 1.0$. The remaining inference parameters were kept constant across experiments, with a maximum generation length of 16,384 tokens, $\text{top}_p = 0.95$, $\text{top}_k = 64$, and reasoning mode enabled.

4.3 Metrics

To evaluate the end-to-end ontology construction, we use the standard metrics defined for the challenge.

4.3.1 Edge F1 (Exact Triple Match)

Let T_g and T_p denote the sets of triples *(subject, predicate, object)* in the gold and predicted graphs, respectively. The **Edge F1** score is the harmonic mean of precision and recall over the full set of triples:

$$S_{\text{Edge F1}} = \text{F1}(T_g, T_p) = 2 \cdot \frac{\text{Precision}(T_g, T_p) \cdot \text{Recall}(T_g, T_p)}{\text{Precision}(T_g, T_p) + \text{Recall}(T_g, T_p)}, \tag{1}$$

where:

$$\text{Precision}(T_g, T_p) = \frac{|T_g \cap T_p|}{|T_p|} \tag{2}$$

$$\text{Recall}(T_g, T_p) = \frac{|T_g \cap T_p|}{|T_g|}. \tag{3}$$

This component rewards exact recovery of triples and penalizes both missing and hallucinated edges.

4.3.2 Neighborhood Similarity (Local Structure)

For each node $v \in V_g \cup V_p$, define its *outgoing neighborhood* as the set of (predicate, object) pairs:

$$N_g(v) = \{(p, o) \mid (v, p, o) \in T_g\}, \quad N_p(v) = \{(p, o) \mid (v, p, o) \in T_p\}. \tag{4}$$

The **Neighborhood Similarity** is the average Jaccard similarity between the gold and predicted neighborhoods for all nodes:

$$S_{\text{Neighborhood}} = \frac{1}{|V_g \cup V_p|} \sum_{v \in V_g \cup V_p} \text{Jaccard}(N_g(v), N_p(v)), \tag{5}$$

where the Jaccard similarity is defined as:

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{6}$$

This measures the structural similarity of local contexts, even when individual triples differ.

4.3.3 Taxonomy Similarity (Hierarchical Structure)

Let $\text{Anc}_g(v)$ and $\text{Anc}_p(v)$ denote the sets of ancestors of node v in the gold and predicted taxonomies, respectively, considering only *is-a* edges. The **Taxonomy Similarity** is the average Jaccard similarity of ancestor sets:

$$S_{\text{Taxonomy}} = \frac{1}{|V_g \cup V_p|} \sum_{v \in V_g \cup V_p} \text{Jaccard}(\text{Anc}_g(v), \text{Anc}_p(v)). \tag{7}$$

This component rewards systems that recover the correct hierarchical positioning of concepts, even if intermediate links vary.

All three components $S_{\text{Edge F1}}, S_{\text{Neighborhood}}, S_{\text{Taxonomy}}$ are bounded in the interval $[0, 1]$. The **Graph Similarity Score** S_{graph} is defined as the average of the three components:

$$S_{\text{graph}} = \frac{S_{\text{Edge F1}} + S_{\text{Neighborhood}} + S_{\text{Taxonomy}}}{3}, \tag{8}$$

where $S_{\text{graph}} \in [0, 1]$. A score of 1 indicates a perfect match, while 0 indicates no overlap.

The metrics are computed in **three variants** to account for different levels of strictness:

- **Exact Match:** Requires a perfect match between triples, neighborhoods, or ancestor sets in the gold and predicted graphs.
- **Fuzzy Match:** Fuzzy Match: Considers a match valid if the aggregate normalised Indel similarity across the elements of the triple (subject, predicate, object) exceeds 0.90.
- **Semantic Match:** Considers a match valid if the product of the cosine similarities between the embeddings of the elements of the triple (subject, predicate, object) exceeds 0.8. The embeddings are generated using a Matryoshka Representation Learning model[3] [13] and normalized over the dataset.

[3] https://huggingface.co/nomic-ai/nomic-embed-text-v1.5

Table 3. *Graph Similarity obtained by the retained prompt variants on the development subset under exact, fuzzy, and semantic matching. The* complete-graph *configuration retains all extracted predicates, whereas the* structure-only *configuration retains only* `is-a` *and* `instance-of`*. The best result for each matching and post-processing configuration is shown in bold.*

Model	Prompt	Structure-only			Complete graph		
		Exact	Fuzzy	Semantic	Exact	Fuzzy	Semantic
Qwen 3.6 35B-A3B	A	0.2794	0.3586	0.3880	0.2637	0.3385	0.3652
	D	0.3188	0.3813	0.4153	0.3020	0.3612	0.3919
Gemma 4 31B	A	0.2874	0.3812	0.3979	0.2521	0.3402	0.3547
	B	0.3044	0.3914	0.4085	0.2702	0.3539	0.3670
	C	0.2965	0.3941	0.4217	0.2694	0.3606	0.3827
	D	0.3745	0.4565	0.4963	0.3344	0.4089	0.4436
	E	0.3893	**0.4707**	0.5061	0.3422	**0.4179**	0.4502
	E ($T = 0$)	**0.3944**	0.4653	**0.5096**	**0.3472**	0.4123	0.4519
	F	0.3787	0.4551	0.5002	0.3464	0.4168	**0.4580**
	G	0.3290	0.4092	0.4407	0.2938	0.3679	0.3950
	H	0.3280	0.3857	0.4150	0.3006	0.3552	0.3809
	I	0.2491	0.2998	0.3208	0.2491	0.2998	0.3208
	J	0.3321	0.4297	0.4583	0.3207	0.4161	0.4443
	K	0.3029	0.3991	0.4367	0.2928	0.3850	0.4223

5 Results and Discussion

5.1 Prompt ablation

Before evaluating the selected configurations on the complete training set, we randomly selected a **100-instance development subset** for the preliminary experiments. Table 3 compares the prompt variants under two post-processing configurations. In the *complete-graph* configuration, all extracted predicates are retained, whereas the *structure-only* configuration keeps only the two dominant structural relations, `is-a` and `instance-of`.

The structure-only configuration improves Graph Similarity for nearly all prompt variants, indicating that additional predicates produced by Text2Graph often introduce relations that are not rewarded by the benchmark[4]. We will retain the structure-only variant for the rest of this paper. The remaining metrics, omitted for readability, exhibit the same overall pattern. Additionally, for Prompts A and D under the same decoding configuration, Gemma 4 31B outperformed Qwen 3.6 35B-A3B; we therefore focused on the former in the subsequent experiments.

We additionally evaluated Prompt E using deterministic decoding ($T = 0$). Although this configuration achieved slightly higher Graph Similarity under exact and semantic matching, while Prompt E with $T = 1$ remained marginally better under fuzzy matching, the observed differences were negligible. A paired Wilcoxon signed-rank test on the per-instance Graph Similarity scores found no statistically significant difference between the two configurations[5]. Based on this result, we retained the default decoding configuration ($T = 1$) for all subsequent experiments.

The prompt ablation highlights several lessons about knowledge graph extraction with LLMs:

[4]Prompt I is unaffected by this post-processing step because it already restricts extraction to the two structural predicates.

[5]Exact match: $p = 0.906$; Fuzzy match: $p = 0.493$; Semantic matching: $p = 0.848$

- **Explicit taxonomy rules are effective.** Clearly distinguishing `rdf:type` from subclass relations and enforcing canonical predicates produce substantial improvements over the baseline.
- **Closed-world constraints help.** Restricting extraction to facts explicitly supported by the input and forbidding inferred intermediate classes reduces spurious relations.
- **Moderate constraints outperform excessive restriction.** Prompt E provides the best overall trade-off, whereas limiting generation to only `is-a` and `instance-of`, as in Prompt I, considerably reduces performance.
- **Filtering is more effective after generation.** Retaining only structural predicates during post-processing performs better than forcing the model to generate only those predicates.
- **Longer prompts are not necessarily better.** The more detailed precision-first Prompts J and K do not outperform the shorter Prompt E, likely because stricter evidence requirements also increase omissions.
- **Examples provide only selective benefits.** The illustrative examples introduced in Prompt F improve some metrics, but do not consistently outperform Prompt E.

Overall, the results favour a moderately constrained, closed-world prompt combined with structural post-processing.

5.2 Full Training and Test Evaluation

***Table 4. Performance of Gemma 4 31B** on the complete training and test sets under exact, fuzzy, and semantic matching. The training-set results compare Prompts D and E after retaining only the structural predicates. The test-set results compare the complete graph produced by Prompt E with its structure-only version. Bold values indicate the best result within each dataset and matching configuration.*

Metric	Matching	Training set (structure-only)		Test set (Prompt E)	
		Prompt D	Prompt E	Complete graph	Structure-only
Edge F1	Exact	0.3485	**0.3578**	0.2885	**0.3256**
	Fuzzy	0.4424	**0.4552**	0.3867	**0.4341**
	Semantic	0.4729	**0.4832**	0.4123	**0.4638**
Neighborhood Similarity	Exact	0.2457	**0.2495**	0.1838	**0.2304**
	Fuzzy	0.3229	**0.3291**	0.2662	**0.3237**
	Semantic	0.3481	**0.3519**	0.2870	**0.3503**
Taxonomy Similarity	Exact	0.4044	**0.4165**	0.3910	**0.3910**
	Fuzzy	0.5397	**0.5542**	0.5723	**0.5724**
	Semantic	0.5765	**0.5901**	**0.6160**	0.6165
Graph Similarity	Exact	0.3329	**0.3413**	0.2878	**0.3157**
	Fuzzy	0.4350	**0.4461**	0.4084	**0.4434**
	Semantic	0.4658	**0.4751**	0.4384	**0.4769**

Table 4 reports two complementary evaluations. On the *training set*, we compare Prompts D and E using the structure-only post-processing selected in Section 5.1 with the goal of verifying whether the prompt improvements observed on the development subset generalize to the full training data. On the *test set*[6], where Prompt E was used for prediction, we instead compare its complete-graph and structure-only outputs to assess whether the structural filtering strategy also generalizes to unseen instances. The training and test columns therefore address different experimental questions and should not be interpreted as a direct comparison between datasets.

[6]The *test set* is not publicly available at the time of writing. All scores related to the test set have been computed by the LLMs4OL challenge organisers, with participating teams having a limited number of submissions.

On the full **training set**, Prompt E consistently outperforms Prompt D across all metrics and matching strategies, confirming that the improvements observed on the development subset generalize to the full dataset. Semantic matching yields the highest scores, followed by fuzzy and exact matching, reflecting the sensitivity of exact evaluation to minor lexical and representational differences. Among the individual metrics, Taxonomy Similarity achieves the strongest results, whereas Neighborhood Similarity remains comparatively lower, suggesting that the system captures hierarchical relations more reliably than the broader local graph structure. Overall, these results support the selection of Prompt E as the best configuration.

On the test set, structural filtering produces the same qualitative effect observed during development: removing non-structural predicates improves all Edge F1 and Neighborhood Similarity variants, while Taxonomy Similarity remains essentially unchanged. Consequently, Graph Similarity increases consistently across all matching strategies, confirming on unseen data the benefit of applying structural filtering after generation. In other words, the conclusions drawn from the training experiments well generalize to unseen data.

We next evaluate whether combining complementary configurations through ensembling can further improve performance.

Table 5. Performance of the ensemble configurations *under exact, fuzzy, and semantic matching. The ensemble retains only triples generated by both constituent configurations. In the training-set experiments, Prompt E with Gemma 4 31B is combined with alternative prompts using the same model or with Qwen 3.6 35B-A3B. The test-set column reports the official results of the selected Gemma 4 31B Prompt E and Prompt F ensemble. Bold values indicate the best training-set result for each metric and matching strategy.*

Metric	Matching	Training set						Test set
		E+D	E+F	E+H	E+I	E+Qwen A	E+Qwen D	E+F
Edge F1	Exact	0.4432	0.4563	0.4071	0.3394	0.4577	**0.4721**	0.3882
	Fuzzy	0.5299	**0.5428**	0.4710	0.4004	0.5252	0.5285	0.5072
	Semantic	0.5664	**0.5710**	0.4928	0.4221	0.5422	0.5499	0.5345
Neighborhood Similarity	Exact	0.3358	0.3478	0.3122	0.2817	0.3766	**0.3885**	0.3078
	Fuzzy	0.4222	0.4305	0.3807	0.3416	0.4446	**0.4500**	0.4235
	Semantic	0.4576	0.4576	0.3984	0.3595	0.4577	**0.4663**	0.4507
Taxonomy Similarity	Exact	**0.4764**	0.4644	0.4159	0.3119	0.4400	0.4394	0.3791
	Fuzzy	**0.5785**	0.5641	0.4911	0.3818	0.5110	0.4990	0.5310
	Semantic	**0.6158**	0.6051	0.5198	0.4065	0.5340	0.5223	0.5633
Graph Similarity	Exact	0.4185	0.4229	0.3784	0.3110	0.4247	**0.4334**	0.3584
	Fuzzy	0.5102	**0.5125**	0.4476	0.3746	0.4936	0.4925	0.4872
	Semantic	**0.5466**	0.5445	0.4703	0.3960	0.5113	0.5128	0.5162

As shown in Table 5, the ensemble results indicate that no single configuration consistently dominates across all evaluation dimensions on the training set. The E+F combination achieves the strongest Edge F1 scores, suggesting a favourable balance between precision and recall at the relation level. In contrast, E+Qwen D performs best in terms of Neighborhood Similarity, indicating a better preservation of local graph structure, while E+D yields the highest Taxonomy Similarity and therefore captures hierarchical relations more accurately. E+F provides the best average performance in terms of Graph Similarity across the three matching strategies. Since Graph Similarity captures multiple complementary aspects of the extracted graph, we selected E+F for the final submission.

Table 6. *Comparison of Prompt E and the ensemble over the 98 inputs for which both configurations produced a non-empty graph.*

Measure	Prompt E	Ensemble
Correct literal matches	519	467
Gold-only triples	555	607
Predicted-only triples	1,392	617
Semantic Edge F1	0.511	0.601
Semantic Neighborhood Similarity	0.366	0.506

Table 7. *Most frequent error patterns for Prompt E and the E+F ensemble over the 98 inputs for which both configurations produced a non-empty graph.*

Error pattern	Prompt E	Ensemble
Extra unmatched `instance-of`	1,023	358
Missed triples without a close equivalent	261	378
Extra unmatched `is-a`	164	83
Lexical mismatches	154	136
Same subject and predicate, different object	63	25
Same endpoints, wrong predicate	51	40
Same object and predicate, different subject	26	28

5.3 Limitations

We further analysed the outputs produced on the development subset by the best-performing prompting configuration, Prompt E, and by the selected E+F ensemble to better understand their respective error patterns and the trade-off between coverage and precision.

Prompt E extracts a larger number of gold relations but also generates substantially more triples that are absent from the reference graph. By retaining only relations shared by both configurations, the ensemble produces smaller and cleaner graphs and better preserves local graph structure, as reflected by its higher Semantic Edge F1 and Neighborhood Similarity. This increased precision, however, comes at the cost of additional omissions: the ensemble misses more gold triples and produces two empty outputs in the complete 100-instance evaluation. Table 6 reports the comparison over the 98 inputs for which both configurations generated a non-empty graph.

Table 7 details the most frequent error categories. The dominant source of over-generation is the introduction of unsupported `instance-of` assertions. Although the ensemble considerably reduces this error, generic type assertions remain its most frequent source of additional triples. Its conservative intersection strategy also increases the number of gold relations for which no close predicted equivalent is found.

Several limitations are shared by both configurations. First, they do not fully resolve the distinction between subclass and instance semantics. Second, canonical entity labels are not always preserved, resulting in mismatches caused by changes in punctuation, casing, spacing, qualifiers, or identifier removal. Third, although the underlying extraction process can generate non-taxonomic relations, our selected challenge configuration filters the output to `is-a` and `instance-of`. This improves benchmark performance given the strong predicate imbalance, but prevents the final system from recovering less frequent axioms such as equivalence, disjointness, overlap, cross-references, and other domain relations.

Finally, predicted-only triples should not automatically be interpreted as incorrect. Some may represent valid assertions that are absent from the reference graph. The

reported results therefore measure alignment with the target ontology rather than unrestricted ontological correctness.

6 Conclusions and Future Work

We presented an LLM-based approach to ontology extraction that combines structured prompting, predicate normalization, structural post-processing, and conservative ensembling. The prompt ablation showed that explicit taxonomy rules and closed-world constraints improve extraction quality, with Prompt E providing the best overall individual configuration. Retaining only the dominant `is-a` and `instance-of` relations further improved Graph Similarity. The selected E+F ensemble achieved official test-set Graph Similarity scores of 0.3584, 0.4872, and 0.5162 under exact, fuzzy, and semantic matching, respectively. Error analysis confirmed that ensembling substantially reduces unsupported relations, although its increased precision comes at the cost of additional omissions.

Future work will investigate softer ensemble strategies that combine confidence or agreement signals without requiring an exact intersection, thereby improving coverage while preserving precision. We also plan to strengthen entity normalization and the distinction between subclass and instance assertions, which remain important sources of error. Finally, we will explore predicate-specific prompts, examples, validation rules, and confidence thresholds to improve the extraction of infrequent non-taxonomic relations.

Data availability statement

The challenge data are available in the LLMs4OL Challenge 2026 repository[1]. The code used for the experiments, together with more detailed results, is available at https://github.com/D2KLab/LettRAGraph-LLMs4OL, except for Text2Graph, which is proprietary software[2].

Author contributions

Pasquale Lisena: Conceptualization, Methodology, Software, Writing – original draft; **Julien Plu:** Methodology, Software, Writing – review & editing; **Oscar Moreno Escobar:** Methodology, Software; **Edouard Trouillez:** Methodology, Software; **Raphael Troncy:** Supervision, Writing – review & editing;

Competing interests

The authors declare that they have no competing interests.

Funding

This work was supported by the French Public Investment Bank (Bpifrance) i-Demo program within the LettRAGraph project (Grant ID DOS0256163/00).

Declaration on Generative AI

During the preparation of this work, the authors used GPT5.5 for grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, and H. M. Abbasi, "A survey of ontology learning techniques and applications," *Database*, vol. 2018, bay101, Jan. 2018, ISSN: 1758-0463. DOI: 10.1093/database/bay101

[2] P. Armary, C. B. El-Vaigh, O. Labbani Narsis, and C. Nicolle, "Ontology learning towards expressiveness: A survey," *Comput. Sci. Rev.*, vol. 56, no. C, May 2025, ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2024.100693

[3] J. Plu, O. Moreno Escobar, E. Trouillez, A. Gapin, and R. Troncy, "A Comprehensive Benchmark for Evaluating LLM-Generated Ontologies," in *23rd International Semantic Web Conference (ISWC), Special Session on Harmonising Generative AI and Semantic Web Technologies (HGAIS)*, CEUR, Ed., Baltimore, USA, 2024.

[4] P. Lisena, J. Plu, O. M. Escobar, E. Trouillez, and R. Troncy, "Pitfalls in AI-Generated Ontologies: Strategies for Detection and Mitigation," in *LLMs4KGOE 2026, International Workshop on LLM-driven Knowledge Graph and Ontology Engineering, co-located with ESWC 2026 (23rd European Semantic Web Conference)*, (May 10–14, 2026), CEUR-WS, Ed., Dubrovnik, Croatia, May 2026.

[5] J. Li, D. Garijo, and M. Poveda-Villalón, "Large Language Models for Ontology Engineering: A Systematic Literature Review," *Semantic Web*, vol. 17, no. 4, 2026. DOI: 10.1177/22104968261465514

[6] L. Peng, P. Yang, Y. Juexiang, and L. Yuangan, "The construction and refined extraction techniques of knowledge graph based on large language models," *Scientific Reports*, vol. 16, no. 1, p. 8104, Feb. 2026, ISSN: 2045-2322. DOI: 10.1038/s41598-026-38066-w

[7] H. Babaei Giglou, J. D'Souza, and S. Auer, "LLMs4OL: Large Language Models for Ontology Learning," in *The Semantic Web – ISWC 2023*, T. R. Payne et al., Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.

[8] H. B. Giglou, J. D'Souza, and S. Auer, "LLMs4OL 2026 Overview: The 3rd Large Language Models for Ontology Learning Challenge," *Open Conference Proceedings*, 2026.

[9] J. Plu et al., "Text2KGBench-LettrIA: A refined benchmark for Text2Graph systems," in *KBC-LM and LM-KBC Challenge (Knowledge Base Construction from Pre-trained Language Models) at ISWC 2025, 24th International Semantic Web Conference, 2-6 November 2025, Nara, Japan*, (Nov. 2–6, 2025), CEUR-WS, Ed., Nara, Japan, Nov. 2025. [Online]. Available: https://ceur-ws.org/Vol-4041/paper3.pdf

[10] T. Ehrhart et al., "ELO-GRAG@EvalLLM 2026: Vers un système HybridRAG combinant graphes de connaissances et texte, évalué sur un corpus de défense en français," in *Proceedings of the EvalLLM 2026 Workshop: Atelier sur l'évaluation des modèles génératifs (LLM), le RAG et challenges*, Nantes, France, Jun. 2026.

[11] Gemma Team et al., "Gemma 4 Technical Report," Tech. Rep., 2026. arXiv: 2607.02770 [cs.CL].

[12] Qwen Team, *Qwen3.6-35B-A3B: Agentic Coding Power, Now Open to All*, Apr. 2026. [Online]. Available: https://qwen.ai/blog?id=qwen3.6-35b-a3b

[13] A. Kusupati et al., "Matryoshka representation learning," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, New Orleans, LA, USA: Curran Associates Inc., 2022, ISBN: 9781713871088.