Journal homepage: https://itce.com.ua/en

Information Technologies and Computer Engineering **Vol. 22, No. 2. 2025**

Article's History: Received: 29.04.2025 Revised: 18.07.2025 Accepted: 28.08.2025

UDC 004.89:004.852 DOI: 10.31649/vitce/2.2025.35

Prompt-guided LLM agent for end-to-end ontology learning

Yaroslav Teplyi*

Postgraduate Student
Lviv Polytechnic National University
79013, 12 Stepana Bandery Str., Lviv, Ukraine
https://orcid.org/0009-0001-5548-5530

Dmytro Dosyn

Doctor of Technical Sciences, Senior Researcher
Lviv Polytechnic National University
79013, 12 Stepana Bandery Str., Lviv, Ukraine
https://orcid.org/0000-0003-4040-4467

Abstract. Turning the vast knowledge of large language models (LLMs) into logical, machine-interpretable ontologies poses a difficult task. Conventional single-prompt strategies often generated ambiguous or mutually inconsistent triples could not be merged safely with reference knowledge bases. The present work therefore aimed to design and empirically verify an entirely automated workflow that converts raw text into schema–compliant RDF/OWL statements without manual intervention. The resulting agent combined four stages – schema discovery, instance extraction, self-repair validation, and ontology alignment – each expressed as structured prompts executed by an LLM. A validator inspected every candidate assertion against the extracted domain-range constraints; any violation triggered an iterative analyse-errors / fix-schema / fix-instances loop until consistency was reached. The workflow was instantiated with three families of LLMs – GPT-4.1-mini, LLaMA-3.3-70b, and Grok-3-mini – chosen to represent high-end proprietary, open-weights, and cost-efficient small models. Quality was measured on the synthetic Measure of Information in Nodes and Edges (MINE) benchmark and on two real scholarly texts: ten English CEUR-WS workshop volumes and ten Ukrainian issues of the Journal of Lviv Polytechnic National University "Information Systems and Networks". On MINE the agent paired with LLaMA-3.3-70b achieved 67.5% fact recall, surpassing the KGGen pipeline (66.07%) while still enforcing schema coherence; GPT-4.1-mini and Grok-3-mini reached 59.8% and 52.4%, respectively. When applied to the bilingual texts, all models reproduced the canonical author–paper–journal relation, proposed up to 39 new classes and 29 new relations, and instantiated more than 800 individuals per dataset with only minor post-repair inconsistencies. Extracted labels remained in English even for Ukrainian inputs. While GPT-4.1-mini and LLaMA-3.3-70b generated broader and valid schema yet most of the schema remained uninitiated, when Grok-3-mini produced concise and fully populated schema with instances. The practical outcome is a pipeline that can be used on digital libraries or domain portals to expand existing knowledge graphs continuously and with minimal human effort, thereby lowering the cost for ontology maintenance and enrichment

Keywords: large language models; knowledge-graph expansion; automated RDF/OWL extraction; schema discovery and alignment; ontology generation

Introduction

Ontology learning from unstructured text has emerged as a critical challenge and opportunity in the age of big data and artificial intelligence (AI). S. Ji *et al.* (2021) explained that an ontology formally represents knowledge as a set of concepts, relations, and rules within a domain, enabling intelligent systems to perform semantic reasoning. F.N. Al-Aswadi *et al.* (2020) pointed out that building ontologies manually is labour-intensive and requires extensive domain expertise, which has driven research into automatic or semi-automatic ontology-learning methods. The modern field has shifted from earlier pattern-based and statistical approaches toward deep learning and language-model techniques. Drawing on the surveys by S. Yuan *et al.* (2022), it was clear that shallow methods – such

Suggested Citation:
Teplyi, Ya., & Dosyn, D. (2025). Prompt-guided LLM agent for end-to-end ontology learning. *Information Technologies and Computer Engineering*, 22(2), 35-47. doi: 10.31649/vitce/2.2025.35

*Corresponding author

as lexical patterns or clustering – are now augmented or replaced by neural models capable of capturing richer linguistic context. Following this evolution, L. Zhong *et al*. (2023) reported a new trend in knowledge-graph construction, where large pretrained language models markedly improve the extraction of concepts and relations from text. Applications such as semantic search, question answering, and recommendation systems benefit from machine-understandable knowledge structures, as demonstrated by G. Karamanolakis *et al*. (2020) and C. Shang *et al.* (2020). They further argued that learning ontologies on the fly facilitates domain adaptation, allowing new ontologies to be created for emerging domains without starting from scratch. Nevertheless, F.N. Al-Aswadi *et al*. (2020) acknowledged persistent challenges: learned ontologies often lack completeness and accuracy because implicit concepts and relations can be hard to detect and erroneous links may be introduced. Integrating extracted ontological knowledge with existing knowledge bases – or aligning it to reference ontologies – remains equally demanding. The advent of transformer-based models and large language models (LLMs) such as BERT and GPT provided powerful text-understanding and generation capabilities, as observed by Y. He *et al.* (2022). Recent studies also emphasise how architectural improvements — for example, optimising memory mechanisms to extend context and support software automation (Sokol, 2025) – contribute to the broader applicability of LLMs in knowledge-intensive tasks. Building on this foundation, B. Chen *et al.* (2023) had begun prompting GPT-style models to generate taxonomy hierarchies or triples, yielding promising, though still preliminary results. Collectively, these developments reinforced the point made by L. Zhong *et al.* (2023): robust ontology-learning solutions are essential for advancing automated knowledge-graph construction and, by extension, the Semantic Web at large.

The article set out to advance automated ontology engineering by validating an LLM-based method that relied on refined prompting and structured-output techniques. To achieve this goal, the study first proposed a workflow that leveraged large language models to extract both schema elements, classes and properties, and instance-level data from unstructured text. It then configured the workflow to reduce the need for manual curation by enforcing structured JSON (JavaScript Object Notation) outputs and iterative self-repair loop, thereby ensuring that the resulting ontological representations remained coherent and ready for integration into existing ontology.

Literature Review

Ontology learning and taxonomy induction from text

Early approaches to ontology learning from text were typically divided into sub-tasks – term extraction, taxonomy induction, and relation extraction – and tackled with pattern-based or statistical methods. As J. Wątróbski (2020) documented, numerous surveys describe these traditional techniques and their applications. K. Belhoucine & M. Mourchid (2018) noted that classic pipelines relied on lexico-syntactic patterns and statistical NLP – such as Hearst-style hypernym patterns or clustering for taxonomy induction – often combined with machine-learning classifiers to extract ontology fragments step by step. Although such pipelines can yield partial ontological structures, they often produce incomplete results when run end-to-end. O. Browarnik & O. Maimon (2015) described the "ontology learning layer cake" problem to describe how independently learned components fail to integrate into a coherent ontology. Consequently, many automated outputs resemble flat knowledge graphs with only simple entity relations and little higher-level class hierarchy, a limitation emphasised by R.M. Bakker *et al*. (2024). They further observed that term-extraction modules – often implemented as named-entity recognisers – successfully identify domain concepts, and relation-extraction techniques can detect some sentence-level links. However, these isolated outputs do not capture the full ontological picture (e.g., distinguishing classes from instances or organising concepts into taxonomies), and large gaps remain in assembling a high-quality ontology, as was underlined by W. Wong *et al*. (2012) and F.N. Al-Aswadi *et al.* (2020). Ultimately, O. Browarnik & O. Maimon (2015) argued that assembling an ontology purely from separately learned layers is inherently difficult, underscoring the continuing need for more integrated approaches.

LLM-based ontology learning

The rapid rise of large language models (LLMs) has opened a fresh research frontier at the intersection of ontology learning and knowledge-graph (KG) construction. Because models such as GPT-3 and GPT-4 absorb vast textual corpora during pre-training, they implicitly encode a broad spectrum of world knowledge that researchers now seek to transform into structured ontologies. One straightforward strategy involves prompting an LLM to generate KG content directly, thereby testing whether the model can handle classic ontology-learning subtasks or even the entire pipeline with minimal human intervention. K. Chen *et al*. (2021) were among the first to show that a pretrained language model can generate taxonomic relations straight from text, learning hypernym hierarchies in a data-driven fashion. Building on that idea, L. Jain & L. Espinosa Anke (2022) demonstrated zero-shot taxonomy induction by probing an LLM with hypernym queries (e.g., "Is X a kind of Y?") to distill concept hierarchies without task-specific training. These early studies established that LLMs do encode substantial semantic knowledge useful for ontology schema construction. Pushing further, H. Babaei Giglou *et al*. (2023) evaluated nine LLMs on three core ontology-learning tasks – term typing (class-vs-instance classification), taxonomy discovery, and non-taxonomic relation extraction under zero-shot prompting. Their results confirmed that LLMs can capture many ontological relations and outperform older rule-based methods on individual subtasks, thanks to the models' rich linguistic representations. Yet, as R.M. Bakker *et al*. (2024)

observed, out-of-the-box LLMs still display inconsistencies and knowledge gaps: they may identify prominent concepts and some subclass relations but overlook subtler links or hallucinate incorrect triples. Taken together, these investigations present LLMs as a promising toolkit for ontology learning, while also highlighting that most evaluations to date focus on isolated subtasks rather than the creation of a complete, integrated ontology.

End-to-end ontology generation

As shown by the 2024 literature analysis, researchers have begun to apply LLMs to generate entire ontologies schema and instancesin an end-to-end fashion. In an in-depth analysis, R.M. Bakker *et al*. (2024) used GPT-4 to extract a complete ontology from domain-specific text: when prompted, the model produced classes, relations, and individual instances. Their study showed that GPT-4 can indeed identify many key classes and entities and even suggest plausible relations, illustrating a holistic grasp of the domain. They also emphasise that, unlike earlier pipelines, the LLM simultaneously considers classes, properties, and instances, thereby building a more integrated ontology instead of isolating each layer. Nonetheless, the fully automatic output still omits certain relations especially object properties linking classes and occasionally introduces erroneous or inconsistent assertions about individuals.

A complementary line of work embeds ontology constraints directly into the prompt. International consortium Monarch Initiative (n.d.) introduced the OntoGPT toolkit, which supplies the LLM with JSON-LD (JavaScript Object Notation for Linked Data) or OWL (Web Ontology Language) templates so that the model fills slots conforming to range and domain restrictions, reducing syntactic post-processing. Comparing prompt engineering with fine-tuning for such structured outputs, B. Chen *et al*. (2023) find that prompt-only approaches are competitive when explicit type constraints are provided. Taken together, the findings of N. Mihindukulasooriya *et al.* (2023) suggested that carefully orchestrated LLM pipelines combining constrained prompting, iterative refinement, and retrieval grounding already outperform classical pipelines on end-to-end ontology metrics. Even so, robustness and explainability remain active research areas before such systems can fully replace expert-driven ontology engineering.

Alignment, integration, and evaluation strategies

B. Mo *et al*. (2025) emphasised that, for an ontology-learning system to be practically useful, the schema elements and instance data it extracts must align with existing knowledge frameworks. To achieve this, their work treated large language models as semantic matchers, prompting them to decide whether a newly extracted term matches or is a subtype of a concept in a target ontology. Complementing this approach, the OntoGPT toolkit (Monarch Initiative, n.d.) embedded ontology-specific constraints directly in the prompt, guiding the LLM to output JSON-LD or OWL structures that conform syntactically and reduce ambiguity.

H. Babaei Giglou *et al.* (2023) further advanced the field through the LLMs4OL Challenge, which offers standardised datasets for tasks such as term typing, taxonomy induction, and relation extraction. Their benchmark suite evaluates not only precision, recall, and F1 scores but also ontology conformance and hallucination rates. Collectively, these efforts and the results they revealed about prompt design, external knowledge grounding, and evaluation protocols show that LLM-based methods have made significant improvements, yet still require refinement before automated ontology learning can fully replace expert-driven engineering.

Materials and Methods

The workflow was implemented in Python; LangChain framework was used to chain template prompts, while preserving intermediate context. Each prompt returned a strictly defined JSON object. The output was validated by Pydantic, which rejected malformed JSON and triggered automatic retry. After every step a validation logic checked domain- and range-consistency; when violations were found, the errors were fed back through prompts until the constraints were satisfied. Three language models (GPT-4.1-mini, LLaMA-3.3-70b, and Grok-3-mini) were chosen as a backend for the developed workflow, the public APIs were used to call the models. The selected models needed to: produce JSON-structured output and support function calls, come in compact versions of state-of-the-art model families, be inexpensive to query, and respond quickly enough for the workflow's many round-trips.

Experiments covered two kinds of material. First, the synthetic MINE benchmark, developed as part of KGGen research by B. Mo *et al*. (2025), (100 articles, 1,500 gold triples) provided an objective ceiling for fact recall. Second, real scholarly text was extracted from ten English CEUR-WS volumes and ten issues of the Journal of Lviv Polytechnic National University "Information Systems and Networks" (Visnyk). HTML pages were fetched, normalised and segmented into token-bounded chunks, and passed to the agent.

Evaluation proceeded on three levels. On MINE it was measured fact-level recall against the gold triples. On the unstructured text, the number of schema elements proposed, those actually instantiated, counts of individuals and property assertions. A sample of final graphs were then manually inspected, checking both the generated triples and the alignment decisions that linked new terms to the backbone ontology. Using both English and Ukrainian data allowed to test whether the agent maintains extraction quality across languages and correctly maps Ukrainian concepts back onto an English schema.

Algorithm. The ontology-learning agent was implemented as an automated, multi-step workflow that consumes raw text and emits a set of RDF/OWL triples ready for insertion into an existing ontology. Figure 1 gives an overview of the data flow, and Algorithm 1 formalises this procedure.

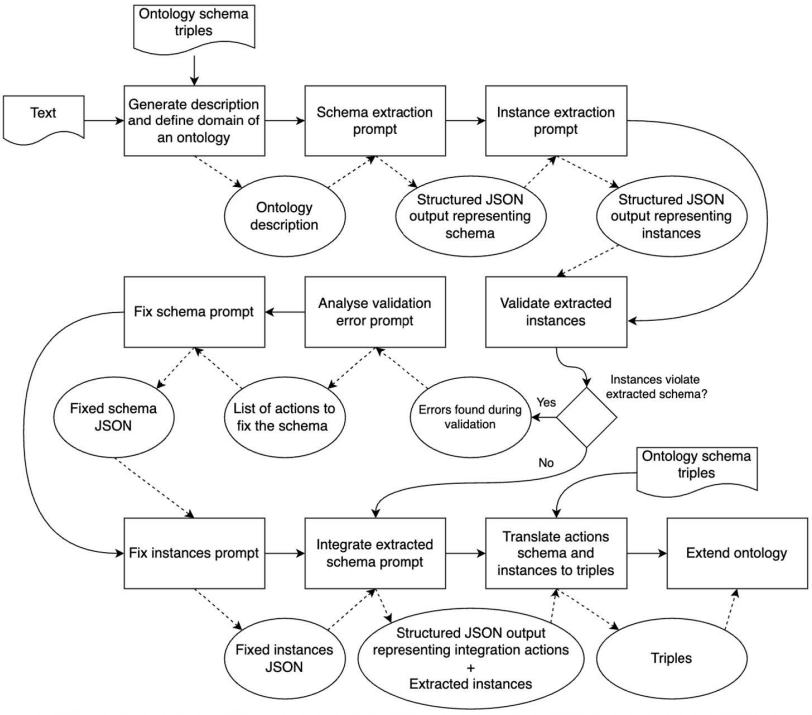

Figure 1. Workflow of the ontology-learning agent
Note: dashed arrows denote data products; solid arrows denote control flow
Source: created by the authors

Ontology-context generation. Given a background ontology , the function Describe extracts a compact natural-language summary of its schema triples. This ontology description is prepended to every subsequent prompt, giving the LLM grounding in domain vocabulary and constraints.

Schema extraction. A schema-extraction prompt instructs the LLM to return {classes, object properties, data properties} in a strict JSON schema.

Instance extraction. For each text chunk, the agent issues an instance-extraction prompt. The result is a draft ABox I whose individuals are asserted to respect the extracted TBox S.

Validation and self-repair. The draft ABox is checked by a deterministic "validate" process that compares every assertion in I against the structural constraints in S. If no violations are found, the workflow proceeds to the next text chunk. Otherwise, the agent enters a “self-repair loop”:

- An “Analyse-errors” prompt receives the validation log and produces a structured list of “repair actions” (e.g. “Verify the existence of “author_name” in instances and create it if necessary”, “Check the “published_in” object property for “paper_1” and ensure its subject exists in instances”).
- Two prompts effect the repairs:

a) fix-schema (updates S when a constraint is missing or malformed);

b) fix-instances (edits or deletes individuals in).

- The modified S and I are re-validated. The loop terminates once “validate” returns “true”.

Ontology integration. The validated schema S and instance set I are aligned with the reference ontology O via two prompts. For every candidate element the LLM outputs an integration action: ignore, new, subClassOf, equivalentClass, or equivalentProperty. The actions are collected in a JSON document A.

Transformation to triples. A transformer converts (S, I, A) into OWL triples. For datatype and class assertions the transformation is straightforward, but object property axioms require additional computations to reconcile the domain and range constraints of newly extracted relations with those already present in O.

For every extracted property that is deemed owl:equivalentProperty to a target in the reference ontology, the agent verifies that its domain and range are compatible with those of the target. Compatibility is determined through an RDFS-subsumption check: an extracted domain d_{ext} is considered compatible with a reference domain d_{ref} when isSubClass(d_{ext}, d_{ref}) holds. If the extracted constraint is a subclass of an existing one, it is safely replaced by the more general superclass; otherwise, the original value is retained. This procedure prevents introduction of overly narrow domain/range declarations.

Finally, each property is serialised as a set of OWL axioms: a rdf:type owl:ObjectProperty declaration for new relations, owl:equivalentProperty links for recognised synonyms, plus the resolved rdfs:domain and rdfs:range statements. Because the transformation stage applies subsumption checks before creating any axiom, it guarantees that

the extended ontology O' preserves the original domain–range semantics and remains logically consistent.

Algorithm 1. Ontology learning agent
Require: ontology O, text T
Ensure: extended ontology
$C \leftarrow Describe(O) \Rightarrow$ context
$S \leftarrow LLM_{schema}(C, T) \Rightarrow$ schema
$I \leftarrow \emptyset$
for each $t \in T$ **do**
  repeat
    $I_s \leftarrow LLM_{inst}(C, S, t)$
  until $Validate(I_s, S) = true$
  $I \leftarrow I \cup I_s \Rightarrow$ instances
end for
$A \leftarrow LLM_{align}(C, S) \Rightarrow$ integration actions
$T \leftarrow Transform(S, I, A)$
$O' \leftarrow O \cup T$
return O'

Complexity notes. Let n_c be the number of text chunks, $|S|$ the size of the provisional schema, and k the average number of validations-repair iterations (empirically $k \leqslant 3$). The dominant cost is $n_c + |S| + k(|S| + |I|) + |S|$ *LLM calls*. In practice k is small because validation targets only coarse inconsistencies (e.g. missing identifiers, range mismatches). By combining (i) schema discovery, (ii) iterative instance extraction with a self-repair loop, and (iii) alignment plus triple generation, the workflow in Figure 1 implements a fully automated, end-to-end ontology learning cycle that needs minimal human intervention.

Results and Discussion

Evaluation on a synthetic benchmark

To obtain a model-agnostic assessment of extraction quality, the MINE benchmark was used, developed as part of KGGen research by B. Mo *et al*. (2025). MINE comprises 100 synthetic articles (~1,000 tokens each) drawn from diverse domains. For every article, 15 reference facts are provided, giving a hard upper bound of 15 × 100 = 1,500 on recall. A system is scored purely on fact-level recall: the proportion of these facts that appear anywhere in the output graph.

Evaluation target. KGGen is designed for surface-level knowledge graph extraction: it prompts the LLM for as many (s, p, o) triples as possible and then clusters lexical variants to maximise recall. The proposed workflow, in contrast, uses the LLM to discover new schema elements (classes, object properties, datatypes) and to validate ABox consistency. Triples that do not satisfy the extracted domain-range constraints are removed.

Absence of ontology context. Because the MINE texts are synthetic, no pre-existing ontology is provided. The workflow therefore skips the ontology-context generation and alignment phases; the integration logic that normally injects additional subclass or equivalence axioms is also never executed. The MINE authors also note that their scoring "rewards density over structure". Thus, while useful as a fact-capture test, MINE is not ideal for measuring the accuracy of an ontology-focused extractor. Figure 2 shows the distribution of recall scores for three backbone LLMs integrated into the workflow.

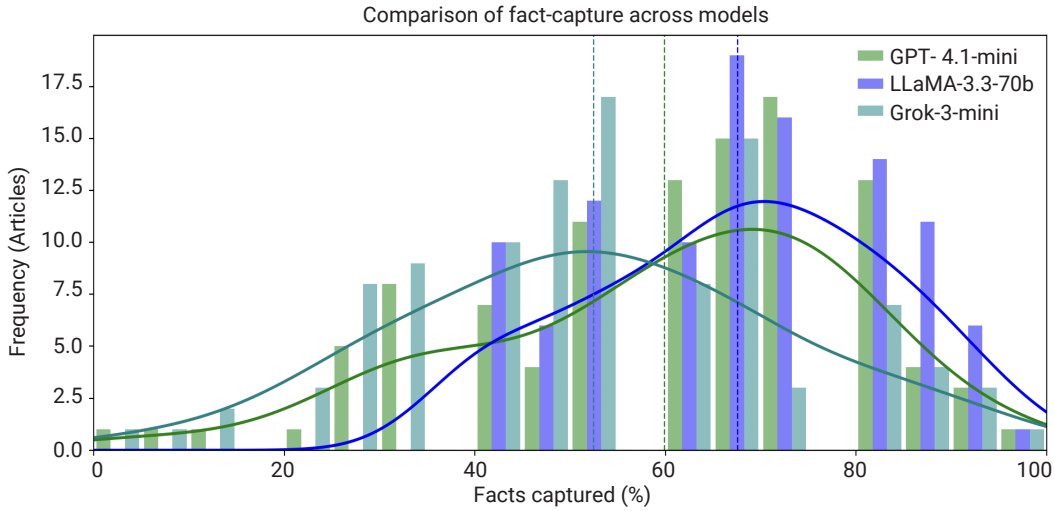

Figure 2. Fact-recall distribution on the MINE benchmark for three backbone models
Note: dotted vertical lines show average performance
Source: created by the authors

With the LLaMA-3.3-70b backend the agent achieved a mean recall of 67.5%, exceeding the 66.07% reported for the KGGen pipeline, and well above other baselines such as GraphRAG (47.80%) and OpenIE (29.84%). The GPT-4.1-mini model also showed good performance scoring (59.8%), while Grok-3-mini trails at 52.4%. These results confirm that the overall workflow, rather than any single LLM, determines ceiling performance: better language models will not always translate directly into higher factual coverage. The results demonstrated that, given a modern LLM backend, the ontology-aware pipeline can match or surpass state-of-the-art fact-capture systems while still enforcing schema consistency.

Case study: Extending a domain ontology from web data

In this case study the ontology-learning workflow was evaluated on scholarly text in two languages (English and Ukrainian) using three language-model backends: GPT-4.1-mini, Grok-3-mini, and LLaMA-3.3-70b. Evaluation was performed on live web sources:

- CEUR-WS – ten random English-language volumes of the CEUR Workshop Proceedings;
- Visnyk – ten Ukrainian-language issues of the "Information Systems and Networks" journal published by Lviv Polytechnic National University.

Both sources provided rich structured information (titles, authors, venues, affiliations, dates, …). The task was to extend an already existent author – publication ontology with schema and instances extracted from each corpus. The comparison focused on each model's multilingual adaptability and on the richness of the ontology it derives. Both schema extraction (TBox) and instance generation (ABox) were measured by counting the classes, properties, individuals, and assertions obtained from the text. For every run three artefacts are recorded:

- the full extracted schema (left panel in Figures 3-5);
- the schema subset actually used when generating instances (right panel in Figures 3-5);
- summary counts of schema elements and instance assertions.

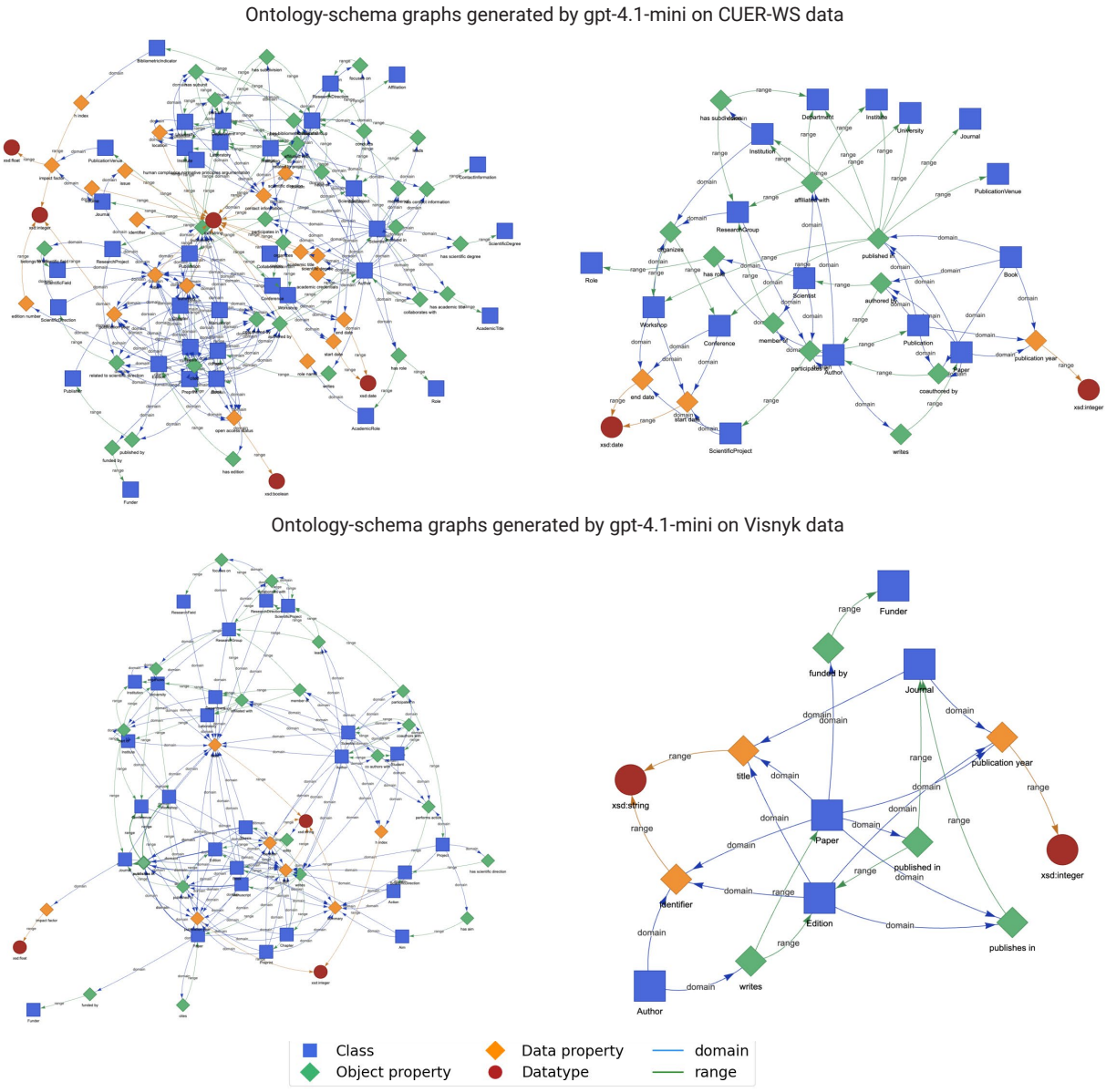

Figure 3. Ontology schema learned from CEUR-WS and Visnyk by GPT-4.1-mini

Source: created by the authors

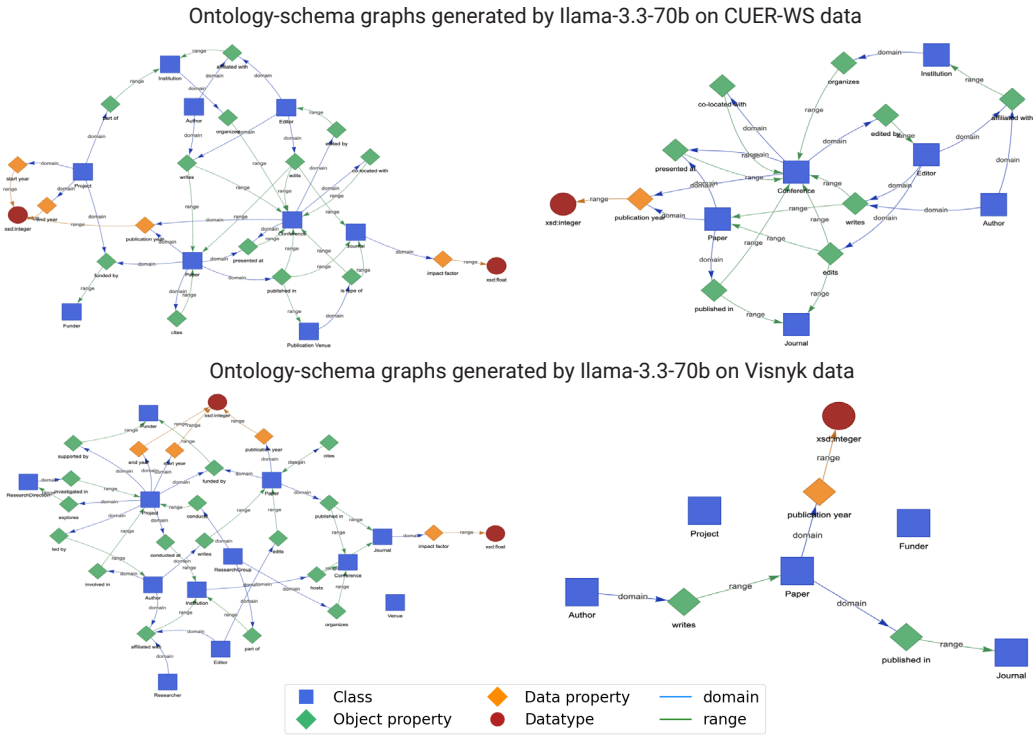

Figure 4. Ontology schema learned from CEUR-WS and Visnyk by LLaMA-3.3-70b
Source: created by the authors

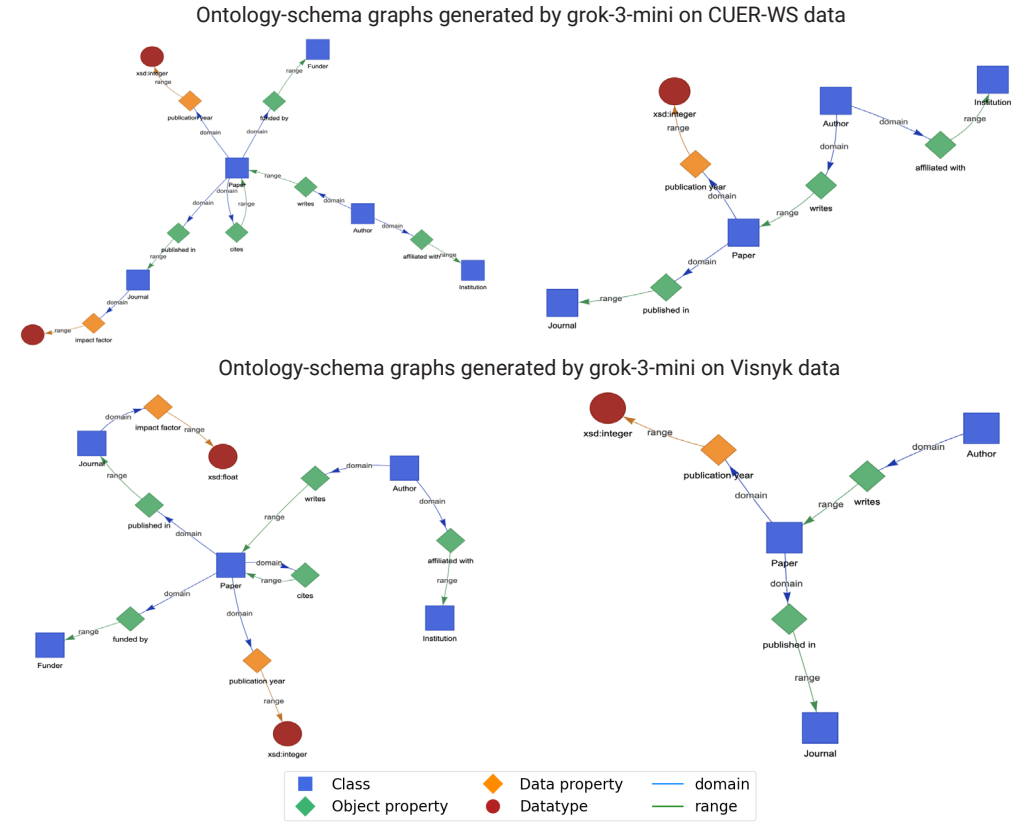

Figure 5. Ontology schema learned from CEUR-WS and Visnyk by Grok-3-mini
Source: created by the authors

Each model was provided with the same initial ontology context (describing key classes like "Author", "Paper", "Conference", "Institution", etc., and their core relations) and then tasked with extracting new ontology elements from the corpora. The prompting and workflow were kept identical for both English and Ukrainian inputs; the ontology context was left untranslated, so an English schema description was used alongside Ukrainian text. A structured ontology was successfully extracted from the Visnyk corpus by all three LLMs without language-specific adjustments, demonstrating the multilingual robustness of the approach. However, the models differed in the scope and detail of what they extracted. The larger models (GPT-4.1-mini and LLaMA-3.3-70b) generally proposed more classes and relationships (a higher "schema richness"), while the smaller Grok-3-mini tended to return a more limited set of concepts. This trend held across both languages, though the gap was more pronounced for the Ukrainian data, where Grok-3-mini likely struggled with some domain terms in Ukrainian.

Across both datasets, GPT-4.1-mini produced the richest schemas (39 classes, 29 object properties on CEUR-WS), followed by LLaMA-3.3-70b, while Grok-3-mini extracted a limited schema (Table 1). Only a subset of these elements was instantiated in the ABox ("Used" columns), highlighting a trade-off: some LLMs hypothesise broader domain structures, but some remain unpopulated in the concrete data. The pattern repeats on the Ukrainian Visnyk corpus, confirming that the difference comes from model capacity rather than language. Despite some differences, it is important to note that all extracted individuals (for all models) passed the validation step. Any anomalies triggered the iterative fix loop, which the models handled in a few iterations. In practice, the quality of the instance data was high for all models; for example, the vast majority of person names extracted were correctly classified as "Author" instances and linked with the right "Institution" or "Paper".

Despite schema size differences, all models extracted a comparable number of individuals (Table 2), confirming that even the smaller model captures the bulk of concrete entities. Larger models do, however, attach richer relational context (object-property counts). Figures 3-5 (bottom panels) demonstrate that every model produced structurally analogous ontologies for the Ukrainian Visnyk corpus without prompt re-engineering. The alignment step reconciled Ukrainian-sourced classes with their English counterparts, mapping, for instance, "Автор" ([avtor]) to "Author".

Table 1. Schema statistics after processing CEUR-WS (English) and Visnyk (Ukrainian)

Category	Source	gpt-4.1-mini		grok-3-mini		llama-3.3-70b	
		Extracted	Used	Extracted	Used	Extracted	Used
Classes	CEUR-WS	39	16	5	4	9	7
	Visnyk	27	5	5	3	12	5
Object properties	CEUR-WS	29	10	5	3	12	8
	Visnyk	19	4	5	2	16	2
Datatype properties	CEUR-WS	19	3	2	1	4	1
	Visnyk	7	3	2	1	4	1

Source: created by the authors

Table 2. Instance statistics on CEUR-WS (English) and Visnyk (Ukrainian)

Category	Source	gpt-4.1-mini	grok-3-mini	llama-3.3-70b
		Count	Count	Count
Individuals	CEUR-WS	797	795	811
	Visnyk	586	615	655
Object properties	CEUR-WS	813	900	984
	Visnyk	654	715	851
Datatype properties	CEUR-WS	18	75	104
	Visnyk	113	161	128

Source: created by the authors

Figure 6 shows a representative diagram of the alignment phase on CEUR-WS. Some elements are mapped as equivalent existing concepts, integrated as subclass extensions or mapped as new, which is mostly applicable for properties, since the backbone ontology doesn't define property hierarchy.

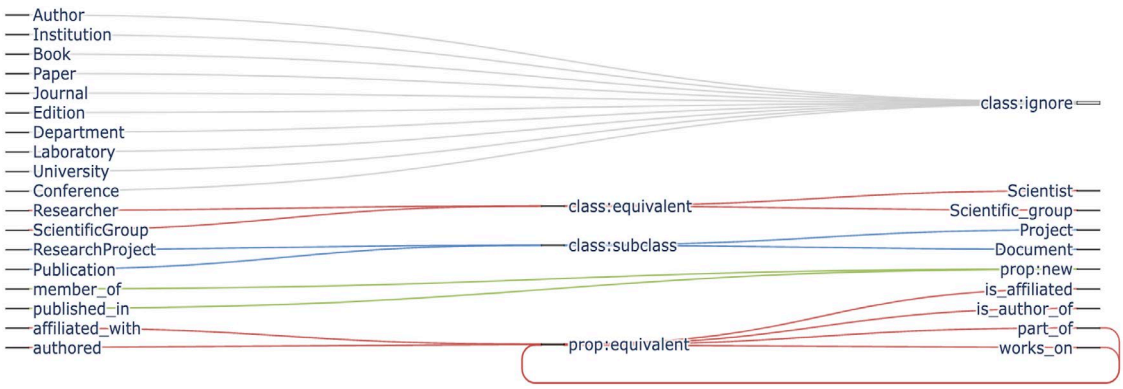

Figure 6. Example alignment-and-integration output (GPT-4.1-mini on CEUR-WS)
Note: colours indicate ignore, equivalent, subclass, and new actions
Source: created by the authors

Most of the concepts were marked as (ignore), as those concepts are already present in backbone ontology. Similar patterns were observed for the other models and the Ukrainian dataset, confirming that the agent can coherently merge LLM-derived knowledge into an existing ontology.

Visual inspection of Figure 7 and manual checks surfaced several limitations:

- Duplicate entities. Chunk-level processing occasionally splits compound works (e.g. names, "Петро Топилко" → "Петро Топи" [petro topylko → petro topy]); consequently, multiple "Author" or "Paper" individuals represent the same real-world entity. A reconciliation layer (string/embedding clustering or rule-based normalisation) is required.
- Unused schema elements. Consistent with Table 1 not every extracted class or property was instantiated; fine-grained concepts and low-level datatypes are underutilised.
- Class misclassification. The issue "Випуск [vypusk] 10, 2021" typed as Journal rather than Edition (where "Випуск [vypusk] 8, 2020" was correct) highlights occasional context loss when semantic cues are distant from the entity mention. Incorporating already extracted instances to agent's context could mitigate this error.
- Missing property links. Some Paper instances lack the "published_in" relation.

Even with the outlined issues, the extracted instances remain a substantive foundation for downstream analytic tasks (e.g., bibliometrics, reviewer recommendation) and can be incrementally refined. In summary, all three LLMs successfully enriched the author–publication ontology from both an English and a Ukrainian corpus. GPT-4.1-mini and LLaMA-3.3-70b proposed broader schemas, while Grok-3-mini produced a concise and fully used the extracted schema. Crucially, the core conceptual backbone was recovered in both languages, attesting to the robustness and multilingual adaptability of the proposed ontology-learning agent.

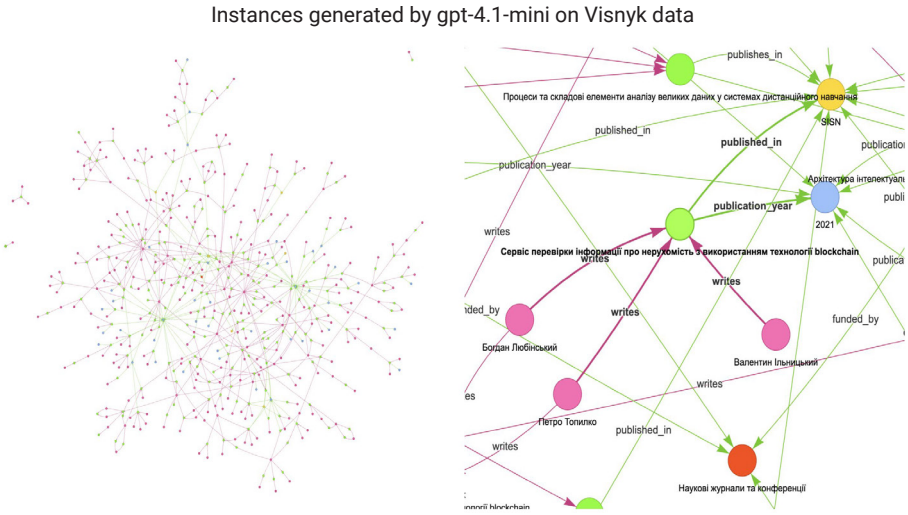

Figure 7. Instances generated by GPT-4.1-mini for the Visnyk corpus
Note: left: complete instance graph (see Table 2), where colors denote inferred classes (paper – green, author – magenta, journal – yellow, data literals – blue). Right: magnified sub-graph illustrating typical link patterns: authors connected to their papers via writes; papers linked to their year of publication by publication_year. Edge direction follows OWL object-property orientation
Source: created by the authors

Compared to other attempts at LLM-based ontology generation, the presented pipeline produces more complete and consistent results. R.M. Bakker *et al*. (2024) demonstrated that GPT-4 can produce an entire ontology from text in a single prompt, but they observed frequent omissions of object properties and occasional erroneous assertions. In their experiments, the LLM often failed to link instances via the proper relationships and sometimes introduced inconsistencies. A similar pattern can be observed in other one-time approaches: M. Funk *et al.* (2023) also reported that LLM-generated ontologies tend to lack certain relational links and contain misclassifications when no iterative feedback is provided. Current results showed a clear improvement in this regard. By incorporating a validation and self-repair loop, the agent ensured that nearly all expected relations were instantiated correctly and domain-range rules were satisfied. Iterative extraction used in this study mitigates common errors noted by the aforementioned authors. This study found far fewer "hallucinatory" facts or missing links in the final ontology, which underscores the value of letting the LLM refine its output with guidance. This outcome was in line with the observations of J.H. Caufield *et al.* (2024), who showed that a recursive extraction strategy reduces errors in LLM-generated knowledge bases. Current work reinforces their finding: structured prompts and iterative verification greatly improve the quality of the extracted ontology compared to single-pass generation.

The generate-validate-repair strategy used in proposed pipeline also resonates with other emerging frameworks. A. Lo *et al.* (2025) proposed a similar end-to-end ontology induction loop, where the LLM's proposed axioms are checked and corrected in successive cycles. Their approach, highlighted that LLMs benefit from explicit error analysis and repair prompts to produce consistent knowledge. The results of current method confirm the viability of iterative prompt-driven ontology building.

Despite these positive comparisons, some limitations of the presented approach reflect existing problems. One issue is the creation of duplicate or fragmented entities due to processing text in chunks. For example, the results showed that the same author might be extracted twice (as separate individuals) if their name was split across two text chunks. This phenomenon is essentially a co-reference resolution problem, well-known in knowledge base construction. Z. Dong *et al.* (2025) address such cross-document co-reference issues by using contextual embeddings to merge duplicate nodes in a knowledge graph. Incorporating a similar entity reconciliation layer could consolidate duplicates in current output as well. Another limitation was that a few schema elements extracted by the LLM remained unused. This suggests the LLM may be over-generalising the schema – a tendency also seen in some taxonomy induction experiments like those by Q. Zeng *et al.* (2024), where iterative prompts produced very granular categories that did not always align with the dataset. While these unused elements do not harm consistency, future work might filter or prioritise schema suggestions based on their frequency or significance in the text, to focus the ontology on well-supported concepts.

Conclusions

In this study, an LLM-driven agent for ontology learning that integrates prompt engineering with structured output enforcement to extract both schema-level and instance-level knowledge from text was presented. The agent first extracts a schema, then populates it with instances, repeatedly checks every assertion against domain–range constraints, and finally reconciles the result with a pre-existing ontology. Evaluations show two complementary strengths. On the synthetic MINE benchmark, the workflow attains a fact-recall score of 67.5% with LLaMA-3.3-70b, overtaking the specialised KGGen pipeline. On two scholarly corpora – English workshop proceedings and Ukrainian journal issues – the same prompts recover the canonical backbone, extend the ontology with dozens of classes and relations, and instantiate more than eight hundred individuals, all without manual intervention or language-specific tuning. Larger LLMs contribute broader schemas, whereas a compact model still produces a fully instantiated but smaller schema.

Generated instances are rich enough for downstream bibliometric studies. Nevertheless, some issues were discovered: chunking occasionally duplicates entities, rare classes and properties are under-utilised, and a handful of classes lacked an important relation links. These issues do not undermine the overall utility of the output, but they do point to the next layer of automation. In sum, the study indicates that carefully constrained LLMs can already assume much of the work traditionally performed by ontology engineers. They can read, propose schemas, populate them, validate their own output, and align the result with an existing knowledge base. By combining recent advances in prompt structuring and structured output, the proposed agent offers a scalable route to semi-automatic, multilingual ontology construction.

Moving forward, there are several areas for improvement. One direction is to incorporate external knowledge or semantic validations (beyond the internal loop) to further minimise any residual errors or hallucinations. Another is to evaluate the approach on more domain-specific benchmarks that reward not just recall but also the quality of the ontology structure (as suggested by emerging challenges like LLMs4OL and Text2KGBench). Finally, prompting techniques are not able to ground the model completely and resulted schema can vary across different model, so finetuning is a promising area of further research. The combination of LLMs with ontology engineering, as exercised by the agent, is as a promising step toward reducing the bottleneck of manual knowledge-base construction and enabling more adaptive, intelligent systems.

Acknowledgements

None.

Funding

The study was not funded.

Conflict of Interest

None.

References

[1] Al-Aswadi, F.N., Chan, H.Y., & Gan, K.H. (2020). Automatic ontology construction from text: A review from shallow to deep learning trend. *Artificial Intelligence Review*, 53, 3901-3928. doi: 10.1007/s10462-019-09782-9.

[2] Babaei Giglou, H., D'Souza, J., & Auer, S. (2023). LLMs4OL: Large language models for ontology learning. In T.R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng & J. Li (Eds.), *The Semantic Web – ISWC 2023* (pp. 408-427). Cham: Springer. doi: 10.1007/978-3-031-47240-4_22.

[3] Bakker, R.M., Di Scala, D.L., & de Boer, M.H. (2024). Ontology learning from text: An analysis on LLM performance. In *Proceedings of the 3rd NLP4KGC international workshop on natural language processing for knowledge graph creation in conjunction with SEMANTiCS 2024 conference* (paper 5). Amsterdam: CEUR-WS.

[4] Belhoucine, K., & Mourchid, M. (2018). A survey of ontology learning from text. In *Proceedings of the twelfth international conference on advances in semantic processing (SEMAPRO 2018)* (pp. 14-21). Athens: IARIA.

[5] Browarnik, O., & Maimon, O. (2015). Ontology learning from text: Why the ontology learning layer cake is not viable. *International Journal of Signs and Semiotic Systems*, 4. doi: 10.4018/IJSSS.2015070101.

[6] Caufield, J.H., Hegde, H., Emonet, V., Harris, N.L., Joachimiak, M.P., Matentzoglu, N., Kim, H., Moxon, S., Reese, J.T., & Haendel, M.A. (2024). Structured prompt interrogation and recursive extraction of semantics (SPIRES): A method for populating knowledge bases using zero-shot learning. *Bioinformatics*, 40, article number btae104. doi: 10.1093/bioinformatics/btae104.

[7] Chen, B., Yi, F., & Varró, D. (2023). Prompting or fine-tuning? A comparative study of large language models for taxonomy construction. In *2023 ACM/IEEE international conference on model driven engineering languages and systems companion (MODELS-C)* (pp. 588-596). Västerås: IEEE. doi: 10.1109/MODELS-C59198.2023.00097.

[8] Chen, K., Lin, K., & Klein, D. (2021). Constructing taxonomies from pretrained language models. In *Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies* (pp. 4687-4700). Stroudsburg: ACL. doi: 10.18653/v1/2021.naacl-main.373.

[9] Dong, Z., Wang, M., Dai, L., Li, J., Liu, X., & Nong, R. (2025). Cross-document contextual coreference resolution in knowledge graphs. *ArXiv*. doi: 10.48550/arXiv.2504.05767.

[10] Funk, M., Hosemann, S., Jung, J.C., & Lutz, C. (2023). Towards ontology construction with language models. In *Proceedings of the KBC-LM'23: Knowledge base construction from pre-trained language models workshop at ISWC 2023* (paper 16). Amsterdam: CEUR-WS.

[11] He, Y., Chen, J., Antonyrajah, D., & Horrocks, I. (2022). BERTMap: A BERT-Based ontology alignment system. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5), 5684-5691. doi: 10.1609/aaai.v36i5.20510.

[12] Jain, L., & Espinosa Anke, L. (2022). Distilling hypernymy relations from language models: On the effectiveness of zero-shot taxonomy induction. In *Proceedings of the 11th joint conference on lexical and computational semantics* (pp. 151-156). Seattle: ACL. doi: 10.18653/v1/2022.starsem-1.13.

[13] Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P.S. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33, 494-514. doi: 10.1109/TNNLS.2021.3070843.

[14] Karamanolakis, G., Ma, J., & Dong, X.L. (2020). Txtract: Taxonomy-aware knowledge extraction for thousands of product categories. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics* (pp. 8489-8502). Stroudsburg: ACL. doi: 10.18653/v1/2020.acl-main.751.

[15] Lo, A., Jiang, A.Q., Li, W., & Jamnik, M. (2025). End-to-end ontology learning with large language models. In *Proceedings of the 38th international conference on neural information processing systems (NIPS '24)* (article number 2767). New York: Red Hook. doi: 10.5555/3737916.3740683.

[16] Mihindukulasooriya, N., Tiwari, S., Enguix, C.F., & Lata, K. (2023). Text2KGBench: A benchmark for ontology-driven knowledge graph generation from text. In T.R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng & J. Li. (Eds.), *The Semantic Web – ISWC 2023* (pp. 247-265). Cham: Springer. doi: 10.1007/978-3-031-47243-5_14.

[17] Mo, B., Yu, K., Kazdan, J., Mpala, P., Yu, L., Cundy, C., Kanatsoulis, C., & Koyejo, S. (2025). KGGen: Extracting knowledge graphs from plain text with language models. *ArXiv*. doi: 10.48550/arXiv.2502.09956.

[18] Monarch Initiative. (n.d.). *OntoGPT: A prompt-based ontology extraction tool*. Retrieved from https://monarch-initiative.github.io/ontogpt.

[19] Shang, C., Dash, S., Chowdhury, M.F.M., Mihindukulasooriya, N., & Gliozzo, A. (2020). Taxonomy construction of unseen domains via graph-based cross-domain knowledge transfer. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics* (pp. 2198-2208). Stroudsburg: ACL. doi: 10.18653/v1/2020.acl-main.199.

[20] Sokol, O. (2025). Optimising productivity and automating software development: Innovative memory system approaches in large language models. *Technologies and Engineering*, 26(1), 36-44. doi: 10.30857/2786-5371.2025.1.3.
[21] Wątróbski, J. (2020). Ontology learning methods from text – an extensive knowledge-based approach. *Procedia Computer Science*, 176, 3356-3368. doi: 10.1016/j.procs.2020.09.061.
[22] Wong, W., Liu, W., & Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Computing Surveys*, 44(4), 1-36. doi: 10.1145/2333112.2333115.
[23] Yuan, S., He, J., Wang, M., Zhou, H., & Ren, Y. (2022). A review for ontology construction from unstructured texts by using deep learning. In *Proceedings of the international conference on internet of things and machine learning (IoTML 2021)* (article number 121741D). Shanghai: SPIE. doi: 10.1117/12.2628713.
[24] Zeng, Q., Bai, Y., Tan, Z., Feng, S., Liang, Z., Zhang, Z., & Jiang, M. (2024). Chain-of-layer: Iteratively prompting large language models for taxonomy induction from limited examples. In *Proceedings of the 33rd ACM international conference on information and knowledge management* (pp. 3093-3102). Arlington: ACM. doi: 10.1145/3627673.3679608.
[25] Zhong, L., Wu, J., Li, Q., Peng, H., & Wu, X. (2023). A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56, article number 94. doi: 10.1145/3618295.

Prompt-керований LLM-агент для комплексного вивчення онтологій

Ярослав Теплий

Аспірант
Національний університет «Львівська політехніка»
79013, вул. Степана Бандери, 12, м. Львів, Україна
https://orcid.org/0009-0001-5548-5530

Дмитро Досин

Доктор технічних наук, старший дослідник
Національний університет «Львівська політехніка»
79013, вул. Степана Бандери, 12, м. Львів, Україна
https://orcid.org/0000-0003-4040-4467

Анотація. Перетворення знань великих мовних моделей (LLM) на логічні, машинно-інтерпретовані онтології є складним завданням. Звичайні стратегії з простими запитами часто породжують неоднозначні або взаємно несумісні триплети, які не можна безпечно об'єднати з існуючими базами знань. У цій роботі розроблено й емпірично перевірено повністю автоматизований робочий процес, що перетворює текст на RDF/OWL-твердження, узгоджені зі схемою, без участі людини. Створений агент об'єднує чотири етапи – виявлення схеми, виявлення екземплярів, самовалідацію з виправленнями та узгодження онтології – кожен реалізовано як структурований запит до LLM. Валідатор перевіряє кожне твердження на дотримання обмежень; будь-яке порушення запускає ітеративний цикл аналіз помилок / виправлення схеми / виправлення екземплярів до досягнення узгодженості. Робочий процес було протестовано з трьома родинами LLM – GPT-4.1-mini, LLaMA-3.3-70b та Grok-3-mini, які представляють висококласні пропрієтарні, відкриті та бюджетні компактні моделі. Якість оцінювалася на синтетичному бенчмарку MINE та на двох реальних наборах текстів: десяти англомовних виданнях CEUR-WS і десяти українських випусках журналу «Вісник Національного університету "Львівська політехніка" "Інформаційні системи та мережі"». На MINE агент із використанням LLaMA-3.3-70b досяг 67,5 % точності, перевершивши KGGen (66,07 %) і водночас забезпечивши узгодженість схеми; GPT-4.1-mini та Grok-3-mini набрали 59,8 % і 52,4 % відповідно. Під час обробки текстів двома мовами усі моделі відтворили відношення автор – публікація – журнал, запропонували до 39 нових класів і 29 нових відношень, а також створили понад 800 індивідів у кожному наборі з лише незначними неузгодженостями. Вилучена схема залишалися англійською навіть для українських ввідних даних. Хоча GPT-4.1-mini та LLaMA-3.3-70b генерували ширші та коректні схеми, проте більшість концептів залишилися без індивідів, а Grok-3-mini запропонувала компактну та повністю заповнену схему з індивідами. Практичним результатом є робочий процес, який можна застосовувати до цифрових бібліотек або галузевих порталів, для автоматизованого розширення наявні графи знань із мінімальними людськими витратами, що знижує вартість для підтримки та збагачення існуючих онтологій

Ключові слова: великі мовні моделі; розширення графів знань; автоматизоване виявлення RDF/OWL; виявлення та узгодження схем; генерування онтологій