Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2025) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

29th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2025)

CoA-Text2OWL: Enhancing Ontology Learning with Chain-of-Agents Framework

Hussam Ghanem[a,*], Samir Jabbar[a], Christophe Cruz[a]

[a] UniversitÃ© Bourgogne Europe, CNRS, Laboratoire Interdisciplinaire Carnot de Bourgogne ICB UMR 6303, 21000 Dijon, France

Abstract

Ontology learning from unstructured text remains a complex challenge, particularly for large and intricate textual sources. This paper introduces CoA-Text2OWL, a multi-agent framework that leverages Large Language Models (LLMs) within a Chain-of-Agents to improve ontology generation. Unlike traditional single-LLM approaches, CoA-Text2OWL distributes the task across multiple worker agents, each processing a chunk of the input text, while a manager agent synthesizes their outputs into a coherent ontology. We evaluate our approach against a baseline single-LLM-based Text2OWL method, demonstrating improvements in object property extraction and ontology completeness. However, challenges remain in preserving hierarchical structures. Our results highlight the potential of multi-agent AI for ontology learning and suggest future enhancements, including specialized agent roles for term extraction, classification, and validation. We further validate CoA-Text2OWL by applying it to construct ontologies from real-world TRACES data related to urban systems in Geneva, achieving strong semantic alignment with source documents. This research contributes to the evolving field of LLM-powered multi-agent systems and their application in knowledge representation.

Keywords: Ontology Learning, Agentic AI, Text2OWL, Large Leanguage Models ;

1. Introduction

Ontology learning (OL), the automatic construction of ontologies from unstructured text, has traditionally relied on linguistic, statistical, and logic-based methods [2, 7]. The rise of large language models (LLMs) has introduced new possibilities for automating this process, leveraging their ability to understand language, reason, and extract relevant information [7, 3]. Simultaneously, Agentic AI has emerged as a paradigm focused on creating autonomous systems capable of complex decision-making with minimal human oversight [1]. These systems enhance adaptability and efficiency but also introduce ethical concerns related to accountability and societal impact [5, 8]. A promising application of LLMs in OL is Text2OWL, which converts textual data into formal ontologies using the Web Ontology

* Corresponding author
E-mail address: hussam.ghanem@u-bourgogne.fr

Language (OWL) [7]. However, challenges arise with longer, more complex texts, where a single LLM may struggle with coherence and completeness. To address these limitations, the Chain-of-Agents (CoA) framework has been introduced, significantly improving long-context tasks [9]. CoA employs multiple "worker" agents to analyze different text segments, with a "manager" agent synthesizing their findings into a cohesive and comprehensive output.

In this paper, we propose a novel approach that integrates LLMs with the CoA framework to enhance OL. By applying CoA to the LLM-based Text2OWL methodology, we aim to improve coherence and completeness in automatically generated ontologies, particularly when dealing with extensive and intricate textual sources. We hypothesize that breaking down the ontology generation process into smaller, manageable stepsâ€"facilitated by multiple collaborating LLM agentsâ€"can yield more accurate and complete ontologies, especially for complex or lengthy textual inputs.

The key contributions of this paper are as follows. We propose a novel CoA-based architecture for LLM Text2OWL ontology generation, leveraging the principles of Agentic AI. We evaluate the performance of our approach on pizza domain ontology dataset used in [7], comparing it to existing single-LLM OL method [7]. We analyze the advantages and disadvantages of using CoA in this context, considering both performance metrics and ethical implications. We provide insights into the design of effective agent communication strategies for OL, contributing to the broader field of LLM-based multi-agent systems. We apply CoA on TRACES data, adopting a reversed evaluation approach to evaluate the generated ontology. The ANR PRCI TRACES project aims to enable the modelling and analysis of the environmental trajectories of territories. The use case is related to the extension of the tram lines in Genevia.

This paper is structured as follows. Section 2 presents a comprehensive review of related work in Agentic AI, Multi-Agent Systems, and both traditional and LLM-based OL approaches. In Section 3, we detail the proposed CoA-Text2OWL methodology, describing the system architecture, agent prompts, and the overall workflow. Section 4 presents the experimental setup, datasets, evaluation metrics, the results obtained, and a discussion of performance, limitations, and challenges. Section 5 then show the application of CoA on TRACES data. Finally, Section 7 concludes the paper with a summary and an outline of future research directions.

2. Related Work

This section provides an overview of related research in two key areas: 1. Agentic AI and Multi-Agent Systems, and 2. Ontology Learning based on traditional and LLM-based approaches.

The field of Agentic AI is rapidly evolving, focusing on creating autonomous systems capable of pursuing long-term goals, making decisions, and executing complex workflows [5]. Agentic AI is distinct from traditional intelligent agents and multi-agent systems due to its ability to tackle open-ended tasks and proactively generate novel solutions [5]. LLM-based multi-agent systems offer advanced capabilities compared to single LLM-powered agents, as these systems can specialize LLMs into various distinct agents with different capabilities, enabling interactions among diverse agents to simulate complex real-world environments effectively [4]. The CoA approach [9] aligns with the cooperative communication paradigm discussed by [4], where agents work together toward shared objectives by exchanging information. In CoA, worker agents, which they are parallel and independent, pass information and insights to a manager agent who synthesizes the final output. The rise of agentic AI presents both opportunities and challenges [5]. As AI systems take on roles with greater independence, it becomes crucial to address issues related to liability attribution, informed consent, and accountability [5]. Furthermore, the potential for emergent self-regulation within networks of agentic AI raises important questions about aligning these norms with societal values and ensuring transparency and accountability [5]. Our work builds upon the principles of agentic AI by leveraging multiple agents to perform OL tasks collaboratively, aiming to improve accuracy and interpretability. This research also aligns with efforts to specialize LLMs into agents with distinct capabilities and enable interactions to simulate complex environments, as noted by [4].

Ontology learning is the process of automatically constructing ontologies from data [2]. Traditional approaches to OL have involved a combination of techniques from natural language processing, machine learning, and knowledge representation [2]. These techniques are often applied to sub-tasks such as term extraction, concept identification, relationship discovery, and axiom formulation [2]. With the advent of LLMs, new approaches to OL have emerged [7]. LLMs have been explored for various OL tasks, including term typing [3], relationship extraction [3], and axiom generation [7, 3]. One particularly relevant approach is direct ontology generation, where an LLM is prompted to directly generate an ontology from text [7]. This is also called Text2OWL, where textual information will be translated to formal OWL ontology [7]. [7] investigated the application of LLMs for unsupervised OL, comparing different LLM-based pipelines to traditional methods. Their study highlighted the potential of LLMs for automating aspects of OL, but also identified limitations related to coherence and completeness. They evaluated three OL pipelines: LLM Text2OWL, OLAF LLM (using only LLM-based components), and OLAF no-LLM (using only non-LLM-based components) [7].

Our work builds upon these efforts by introducing a multi-agent approach to the LLM Text2OWL pipeline. By leveraging the CoA framework [9], we aim to overcome the limitations of single-LLM approaches and improve the accuracy and completeness of automatically generated ontologies. This aligns with current research to use LLM-based multi-agents to solve various tasks, such as software development, multi-robot systems, society simulation, policy simulation, and game simulation [4].

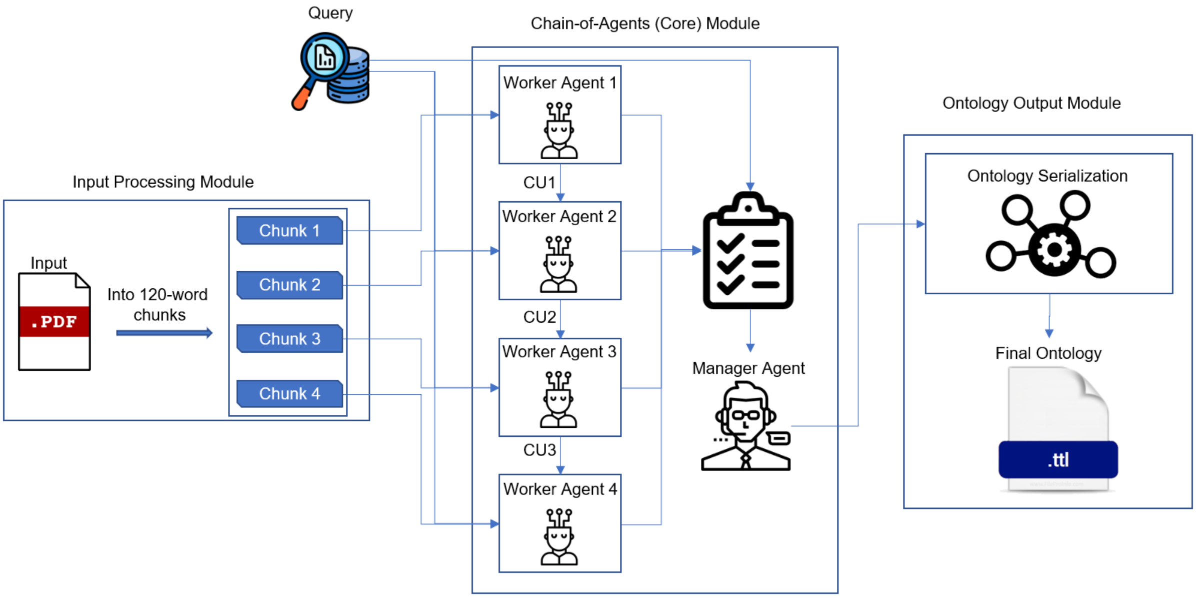

Fig. 1. CoA-Text2OWL Architecture

3. Methodology: CoA-Text2OWL

Our CoA-Text2OWL framework implements a novel approach to ontology generation from PDF documents (textual documents) using CoA architecture [9]. This section details the system's components and workflow.

The CoA-Text2OWL system comprises three main components (fig. 1):

1. **Input Processing Module**: Handles PDF parsing and text segmentation.
2. **Chain-of-Agents Core**: Consists of worker agents and a manager agent for distributed ontology construction.
3. **Output Generation Module**: Responsible for final ontology synthesis and serialization.

Our system utilizes carefully crafted prompts, drawing inspiration from the prompt format used by [7], to guide the behavior of both worker and manager agents. However, the content of the prompts was specifically designed and adapted to our CoA architecture. For worker agents, the prompt emphasizes their role in ontology construction and extracting key ontological elements from a specific text chunk:

Prompt 1: You are a helpful assistant in building an ontology. You are fluent in the W3C Semantic Web stack and RDF, RDFS, and OWL languages. Use the given text to construct an OWL ontology in the Turtle format. Use this namespace: http://example.org/example#. Return only the turtle file. Extract all the possible classes and subclasses.

This prompt is designed to focus workers on extracting classes and subclasses within the confines of their assigned chunk, while adhering to specific syntax guidelines for OWL ontologies in Turtle format. The manager agent's prompt, on the other hand, emphasizes its role in integration and coherence, ensuring that the final ontology is a unified and consistent representation of the input text:

Prompt 2: You are an ontology learning expert, and your goal is to integrate some sub-ontologies into a coherent ontology while ensuring no redundancy. Your response should be structured, modular, and ready for further reasoning and inference.

These prompts are crucial in guiding the agents' behavior, ensuring that worker agents focus on extraction and initial structuring, while the manager agent concentrates on integration and overall coherence of the final ontology. While inspired by the format of [7], the content is uniquely tailored to leverage the strengths of our CoA approach.

Our approach transforms PDF documents into formal ontologies through a sequential Chain-of-Agents system that extracts, processes, and synthesizes textual information across distributed workers before final integration. The system begins by ingesting a PDF document using PyMuPDF[1]. The extracted text undergoes a segmentation process, dividing it into chunks. We choose a size that balance context preservation with the optimal input capacity for the language model.

The core of our system leverages a sequential CoA approach. The agents communicate through a context-preserving mechanism, allowing for the accumulation and propagation of knowledge across text chunks. Each worker agent can potentially build upon the context provided by previous agents, enhancing the coherence of the generated ontology.

1. **The Worker Agents**: Multiple instances of WorkerAgent, powered by the open-mistral-7b model[2], process individual text chunks. Each worker agent receives a chunk of text, the user's query (e.g., "Use the given text to construct an OWL ontology in the Turtle format"), and the output from the previous agent (initially None). Workers generate partial ontologies in Turtle format based on their assigned chunk.
2. **The Manager Agent**: A single ManagerAgent, also using the open-mistral-7b model, synthesizes the outputs from all worker agents into a coherent final ontology.

The manager agent performs the crucial task of integrating partial ontologies from worker agents. This process involves 1/ Merging consistent concepts (classes) and relationships, 2/ Resolving potential conflicts between partial ontologies, 3/ Ensuring overall structural integrity of the final ontology. The final ontology is serialized into a Turtle (.ttl) file, adhering to standard RDF and OWL conventions. This methodology leverages the distributed processing of text chunks to generate a comprehensive ontology, with a focus on context preservation between chunks. The sequential processing of chunks allows for the construction of a coherent and potentially more comprehensive ontology than would be possible with isolated chunk processing.

[1] PyMuPDF: `https://pymupdf.readthedocs.io/en/latest/`
[2] Mistral-7b: `https://mistral.ai/fr/news/announcing-mistral-7b`

4. Experiments

Our experimental evaluation consists of three key components. In the Datasets section, we detail our use of pizza domain ontology texts which we segment into balanced chunks for processing. The Baseline Methods section introduces our comparative approach against a single-LLM direct Text2OWL method without multi-agent coordination. Finally, the Evaluation Metrics section outlines our assessment framework using established ontology learning metrics that measure completeness, conciseness, and correctness to quantify the effectiveness of our approach in extracting structured knowledge.

4.1. Datasets

Our experimental evaluation leveraged the pizza domain ontology dataset used in [7], where textual summaries of a pizza ontology were generated using LLMs. It comprises 10 structured textual descriptions covering diverse concepts related to pizza, including varieties, ingredient classifications, preparation methods, and compositional constraints. These descriptions inherently encode hierarchical relationships, taxonomic structures, and complex constraints, making them highly suitable for evaluating ontology extraction methods. To adapt the dataset for OL with our CoA-Text2OWL framework, we applied a segmentation process that divided texts into balanced chunks, ensuring compatibility with the agent-based architecture. No additional normalization or preprocessing (e.g., tokenization or linguistic modifications) was appliedâ€”only chunking was used to structure the input, maintaining a realistic and unaltered processing scenario. The ontology used to generate the textual descriptions via LLMs is summarized in the first column (Pizza Ontology) in Table 2.

4.2. Baseline Methods

To evaluate the performance of our CoA-Text2OWL approach, we conducted a comparative analysis against a baseline that follows the methodology presented by [7], where ontology extraction is performed using a single LLM instance without multi-agent coordination. This method, commonly referred to as direct Text2OWL, relies on a Mistral-7B model to generate OWL ontologies from text. The model is prompted to extract concepts, object properties, and hierarchical structures in a single pass. While this approach benefits from the expressive capabilities of large language models, it suffers from limitations in coherence, particularly when processing long texts. Without iterative refinement or structured communication between processing stages, the generated ontology may lack consistency and completeness. By this comparison, we aim to demonstrate how our CoA-Text2OWL framework improves OL by enabling multi-agent collaboration, allowing agents to iteratively refine and structure the extracted knowledge.

4.3. Evaluation Metrics

To assess the quality of the generated ontologies, we employ a set of well-established ontology evaluation metrics. These metrics allow us to measure how well the learned ontology aligns with a reference ontology by examining its completeness, conciseness, and correctness. These three criteria are commonly used in OL to ensure that the extracted knowledge is both meaningful and accurate [6].

Conciseness measures the extent to which the generated ontology avoids unnecessary or irrelevant elements. An ontology is considered concise if it does not contain extraneous concepts, relationships, or axioms that are not relevant to the domain. We compute conciseness as the proportion of elements in the generated ontology that are also present in the reference ontology, normalized by the total number of elements in the generated ontology where O_L represents the learned ontology and O_R represents the reference ontology.

$$Conciseness = \frac{|O_L \cap O_R|}{|O_L|}$$

Completeness evaluates whether the generated ontology sufficiently covers the domain knowledge present in the reference ontology. An ontology is considered complete if it includes all the key elements necessary for representing the target domain. Completeness is calculated as the proportion of elements in the reference ontology that are correctly captured in the learned ontology. A high completeness score indicates that the OL approach has successfully captured most of the domain-relevant concepts and relationships.

$$Completeness = \frac{|O_L \cap O_R|}{|O_R|}$$

Correctness is a composite metric that ensures that the generated ontology is both concise and complete. It is computed as the harmonic mean (F1-score) of conciseness and completeness. This metric provides a balanced assessment by penalizing cases where an ontology is either overly verbose (low conciseness) or missing key domain knowledge (low completeness).

$$Correctness = 2 \times \frac{Conciseness \times Completeness}{Conciseness + Completeness}$$

These metrics allow us to quantify the effectiveness of our approach in generating ontologies that are both precise and comprehensive. By comparing the CoA-Text2OWL framework against baseline methods using these measures, we ensure a rigorous evaluation of the strengths and weaknesses of each approach in the OL process.

4.4. Experimental Design

To evaluate the performance of our CoA-Text2OWL framework, we conducted controlled experiments with varying chunk sizes and agent chain lengths. These variations allowed us to analyze the impact of segmentation granularity and multi-agent collaboration on OL. We tested chain lengths from 2 to 5 worker agents, where each agent processed a text chunk before passing its output to the next. Chunk sizes ranged from 100 to 250 words, balancing granularity and context retention. The Mistral-7B model was used for all agent interactions with temperature = 0 to ensure deterministic outputs. The best-performing configuration, maximizing accuracy and coherence in ontology extraction, is summarized in Table 1.

Table 1. Best-performing experimental configuration for CoA-Text2OWL.

Parameter	**Value**
Chunk Size	150
Chunks Number	3
Number of Worker Agents	3
Model	Mistral-7B
Temperature	0

5. Results and discussion

To evaluate the generated ontologies, we first conduct a quantitative analysis. Table 2 displays the counts of OWL named classes, object properties, named individuals, and `RDFS subClassOf` tuples for the reference Pizza Ontology, our CoA-Text2OWL approach, and the baseline Text2OWL method using Mistral-7B. This comparison offers insight into the structural complexity and conceptual richness of each ontology. Following this structural comparison, we assess the quality of the generated ontologies using the standard ontology evaluation metrics: conciseness, completeness, and correctness. These metrics are computed based on the alignment of extracted concepts, object properties, and hierarchical relations (subClassOf) with the reference ontology (Table 3).

Table 2. Comparison of OWL axioms counts for the Pizza Ontology, CoA-Text2OWL, and baseline.

Count	**Pizza Ontology**	**CoA-Text2OWL**	**Baseline**
Named Classes	97	33	23
Object Properties	8	2	0
Named Individuals	5	0	0
Indiv. (ObjProps)	5	0	0
SubClassOf Tuples	141	32	23

Table 3. Comparison of OL performance between CoA-Text2OWL and the baseline method using Mistral-7B.

Metrics	**CoA-Text2OWL**	**Baseline [7]**
Classes conciseness	0.84	**1.00**
Classes completeness	**0.29**	0.25
Classes correctness	**0.43**	0.41
Object properties conciseness	**0.25**	0.00
Object properties completeness	**1.00**	0.00
Object properties correctness	**0.40**	0.00
SubClassOf pairs conciseness	0.12	**0.56**
SubClassOf pairs completeness	0.01	**0.05**
SubClassOf pairs correctness	0.02	**0.09**

The results in Table 3 demonstrate that CoA-Text2OWL significantly outperforms the baseline in object property extraction, achieving complete retrieval of relational structures, while the baseline fails. For class-level extraction, CoA-Text2OWL shows better precision by avoiding unnecessary elements, though its domain concept coverage remains slightly higher than the baseline. In hierarchical relations (subClassOf), the baseline method performs better, suggesting that CoA-Text2OWL struggles with dependency extraction in ontology structures. While it achieves improved correctness for class extraction, its performance is lower in maintaining subclass hierarchies. CoA-Text2OWL excels in precision and relational structure identification but requires further refinement for capturing hierarchical relationships. These findings highlight the advantages of multi-agent OL while pointing to areas for optimization.

The evaluation of CoA-Text2OWL against the baseline Text2OWL using Mistral-7B reveals key insights into ontology extraction performance. The most significant improvement is observed in object property extraction, where CoA-Text2OWL achieves 100% completeness, whereas the baseline fails to extract any object properties. This suggests that the multi-agent processing framework is better suited for capturing relational structures in text. For class-level metrics, CoA-Text2OWL exhibits higher conciseness, reducing unnecessary elements while maintaining a completeness score slightly above the baseline. However, subclass relationships remain a challenge, with the baseline method achieving better `RDFS subClassOf` conciseness and completeness. This indicates that while CoA-Text2OWL enhances certain aspects of OL, further refinements are needed to improve taxonomic structure extraction. In summary, the primary advantage of CoA-Text2OWL lies in its ability to identify object properties, which the baseline approach fails to extract. However, both methods exhibit a significant reduction in hierarchical relationships, suggesting that OL from text remains a challenging task, particularly for subclass extraction.

Despite its improvements over the baseline, CoA-Text2OWL faces several challenges **1. Hierarchical Structure Loss**: The reduced number of `RDFS subClassOf` relationships suggests that multi-agent processing does not fully preserve hierarchical dependencies, likely due to segmentation and inter-agent communication gaps. **2. Limited Con-**

cept Coverage: While CoA-Text2OWL outperforms the baseline in extracting object properties, it still captures significantly fewer named classes than the reference ontology. Expanding coverage without increasing noise remains an open problem. **3. LLM Stochasticity**: Despite setting `temperature=0` for deterministic outputs, minor variations in generated ontologies were observed across repeated runs. This highlights the inherent non-deterministic nature of LLM-based ontology extraction and suggests that additional stabilization techniques, such as structured verification layers, could be beneficial. **4. Scalability Constraints**: While CoA-Text2OWL demonstrates improvements over direct Text2OWL, its scalability to large, complex domains remains an open question. Future research should explore optimizations for processing efficiency and memory retention to enable broader applicability.

6. Evaluation on TRACES Data

To evaluate the practical relevance of our CoA framework for OL, we applied it to a real-world urban systems scenario in Geneva, Switzerland (See Appendix Appendix A). This use case focused on two complementary knowledge domains. The structure and development of the **tramway network** in Geneva, including lines, vehicles, infrastructure, and spatial nodes. The **public policies** regulating urban mobility, transport planning, sustainability measures, and governance frameworks.

We scraped publicly available documents and structured them into two distinct PDF corpora; a collection of Wikipedia pages and institutional descriptions concerning Geneva tramway lines and a corpus of legislative, strategic, and planning texts covering local mobility policy and infrastructure governance. Each document was used to generate an ontology using the LLM `mistral-large-2411`. The resulting OWL/Turtle ontologies were then converted back into natural language summaries using GPT-4o, enabling semantic comparison with the original documents.

Given the absence of a ground-truth ontology for this domain, we adopted a reverse evaluation approach. We used **multilingual BERT embeddings** to compare the semantic content of the generated ontology summaries with the original source documents. Specifically, we split the source PDFs into approximately 512-token chunks. We computed cosine similarity between each chunk's embedding and the embedding of the ontology summary. We aggregated average, maximum, and minimum similarity scores. And, we supplemented this with traditional metrics: ROUGE-1/2/L and BLEU.

Table 4 summarizes the similarity metrics obtained for each domain. These results suggest that both ontologies captured the core semantics of their respective domains, with slightly higher lexical overlap observed for the policy ontology. The high cosine similarities (above 0.92 in both cases) reflect strong conceptual alignment. The lower ROUGE-2 and BLEU-2 scores are expected, as the ontology summaries use abstracted language rather than surface-form replication. However, the results of BLEU-1 and ROUGE-1 that CoA generate classes, object properities, and subclasses from the ground truth text.

Table 4. Evaluation results for ontology-document semantic similarity.

Ontology	BERTScore	ROUGE-1	ROUGE-2	BLEU-1	BLEU-2
Tramway Network	0.935	0.365	0.059	0.240	0.079
Mobility Policy	0.926	0.392	0.089	0.249	0.098

This experiment confirms that the CoA framework can successfully extract domain-specific ontological knowledge from heterogeneous, unstructured corpora and encode it in a form that remains semantically faithful to the source. The evaluation pipeline based on reverse summarization and embedding comparison offers a practical alternative in scenarios lacking annotated gold standards.

7. Conclusion and Future Work

In this work, we introduced CoA-Text2OWL, a multi-agent framework for ontology learning, and evaluated it against an LLM-based Text2OWL approach. Our results show that CoA-Text2OWL excels in object property extraction, achieving complete identification of relational structures, unlike the baseline. At the class level, it produces a more concise ontology, effectively filtering irrelevant elements. However, its coverage of domain concepts remains only slightly higher than the baseline. A key limitation is in subclass extraction, where the baseline performs better, suggesting challenges in maintaining hierarchical dependencies within the agent-based approach.

While CoA-Text2OWL demonstrates meaningful progress in OL, several challenges remain open for future research. An important direction involves refining the CoA framework to support OL at each stage of the process. By involving it in term extraction, concept classification, relation identification, hierarchy construction, and axiom generation. A promising enhancement is the introduction of task-specific workers within the CoA scheme. Rather than having a homogeneous sequence of agents, we propose an approach where each worker W_i is followed by a dedicated worker W_j that handles a specialized task, such as extracting only object properties, identifying only hierarchical relationships, or validating only class-to-class mappings.

In summary, extending CoA-Text2OWL to a fully modular, multi-stage OL framework with specialized agents for different tasks and dynamic agent coordination presents an exciting research direction. These advancements would enhance hierarchical extraction, relational accuracy, and scalability, ultimately improving the quality of automated ontology generation from text.

8. Acknowledgments

The research project was supported by the Agence Nationale de la Recherche (ANR) under Grant No. ANR-21-CE23-0034 (TRACES) and Grant No ANR-24-LCV1-0008 (LAMAE).

Appendix A. Traces dataset

In this appendix, we provide links to the Trace dataset used in our experiment. Below is a list of the referenced documents:

- 20 minutes. “Extension du tram 15: Maudet r´epond aux craintes des frontaliers.” 20 Minutes, Apr. 2024. https://www.20min.ch/fr/story/genevefrance-voisine-extension-du-tram-15-maudet-repond-aux-craintes-des-frontaliers-103092
- Département des infrastructures, République et canton de Genève. “MOBILITÉ: Des actions fortes pour respecter les objectifs du plan climat cantonal.” June 2022. https://www.ge.ch/document/28906/telecharger.
- Grand conseil de la République et canton de Genève. ”Loi modifiant la loi sur le réseau des transports publics (LRTP) (12553).” June 2020. https://ge.ch/grandconseil/data/loisvotee/L12553.pdf.
- INGEROP – UGUET – FOLIA – CITEC – T-INGENIERIE. ”DOSSIER D’ENQUETE PUBLIQUE PREALABLE A LA DECLARATION D’UTILITE PUBLIQUE.” Aug. 2016. https://web.archive.org/web/20160818100906/http://www.annemasse-agglo.
- Le Tram Piétonnisation - Annemasse. ”Info travaux.” https://www.tram-pietonnisation.fr/info-travaux/.
- Le dauphiné libéré. ”En images. Tram et piétonnisation : le centre-ville d’Annemasse quasi méconnaissable, et c’est pas fini!” Feb. 2024. https://www.ledauphine.com/economie/2024/02/17/tram-et-pietonnisation-le-centre-ville-d-annemasse-quasi-meconnaissable-et-c-est-pas-fini.
- Léman bleu tv. ”L’extension du tram 15 inaugurée.” Dec. 2023. https://www.lemanbleu.ch/fr/Actualites/Geneve/L-extension-du-tram-15-inauguree.html.
- Munafò, Sébastien, Derek Christie, Stéphanie Vincent-Geslin, and Vincent Kaufmann. ”Typologie et évolution des logiques de choix modal chez les actifs motorisés urbains.” Genève, Lausanne, Switzerland, Nov. 2012. https://infoscience.epfl.ch/record/186362/files/Rapport%20choix%20modal.pdf.

- Pays de Gex agglo. ”Prolongement du tramway à Ferney-Voltaire depuis Genève.” https://www.paysdegexagglo.fr/ficheaction/96/8834-prolongement-du-tramway-a-ferney-voltaire-depuis-geneve.htm.
- République et canton de Genève. ”SIL Genève PUBLIC - Loi sur le réseau des transports publics (LRTP) H 1 50.” Mar. 1988. https://silgeneve.ch/legis/index.aspx.
- République et canton de Genève. ”MISE EN SERVICE INTÉGRALE DU LÉMAN EXPRESS : MESURES D'ACCOMPAGNEMENT.” Dec. 2018. https://www.ge.ch/document/13455/telecharger.
- “Ligne 13 du tramway de Genève.” Wikipédia, Dec. 2023. https://fr.wikipedia.org/w/index.php?title=Ligne_13_du_tramway_de_Gen%C3%A8ve&oldid=210179936.
- “Ligne 14 du tramway de Genève.” Wikipédia, Dec. 2023. https://fr.wikipedia.org/w/index.php?title=Ligne_14_du_tramway_de_Gen%C3%A8ve&oldid=210870298.
- “Ligne 17 du tramway de Genève.” Wikipédia, Dec. 2023. https://fr.wikipedia.org/w/index.php?title=Ligne_17_du_tramway_de_Gen%C3%A8ve&oldid=210212545.
- “Ligne 15 du tramway de Genève.” Wikipédia, Jan. 2024. https://fr.wikipedia.org/w/index.php?title=Ligne_15_du_tramway_de_Gen%C3%A8ve&oldid=211900528.
- “Ligne 18 du tramway de Genève.” Wikipédia, Jan. 2024. https://fr.wikipedia.org/w/index.php?title=Ligne_18_du_tramway_de_Gen%C3%A8ve&oldid=211392161.

References

[1] Acharya, D.B., Kuppan, K., Divya, B., 2025. Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey. IEEE Access .

[2] Armary, P., El-Vaigh, C.B., Narsis, O.L., Nicolle, C., 2025. Ontology learning towards expressiveness: A survey. Computer Science Review 56, 100693.

[3] Giglou, H.B., D'Souza, J., Sadruddin, S., Auer, S., 2024. Llms4ol 2024 datasets: Toward ontology learning with large language models, in: Open Conference Proceedings, pp. 17–30.

[4] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O., Zhang, X., 2024. Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680 .

[5] Mukherjee, A., Chang, H.H., 2025. Agentic ai: Expanding the algorithmic frontier of creative problem solving. arXiv preprint arXiv:2502.00289 .

[6] Raad, J., Cruz, C., 2015. A survey on ontology evaluation methods, in: International conference on knowledge engineering and ontology development, SciTePress. pp. 179–186.

[7] Schaeffer, M., Sesboüé, M., Charbonnier, L., Delestre, N., Kotowicz, J.P., Zanni-Merk, C., 2024. On the pertinence of llms for ontology learning. NLP4KGC: 3rd International Workshop on Natural Language Processing for Knowledge Graph Creation, September 17, 2024, Amsterdam, Netherlands .

[8] Shavit, Y., Agarwal, S., Brundage, M., Adler, S., O'Keefe, C., Campbell, R., Lee, T., Mishkin, P., Eloundou, T., Hickey, A., et al., 2023. Practices for governing agentic ai systems. Research Paper, OpenAI .

[9] Zhang, Y., Sun, R., Chen, Y., Pfister, T., Zhang, R., Arik, S., 2025. Chain of agents: Large language models collaborating on long-context tasks. Advances in Neural Information Processing Systems 37, 132208–132237.