CoA-Text2OWL: A Chain-of-Agents Framework for Ontology Learning

Hussam Ghanem[1], Samir Jabbar[1], and Christophe Cruz[1]

Université Bourgogne Europe, CNRS, Laboratoire Interdisciplinaire Carnot de Bourgogne ICB UMR 6303, 21000 Dijon, France
hussam.ghanem@u-bourgogne.fr

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. 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, the task of automatically constructing and populating ontologies from unstructured text, has traditionally relied on a combination of linguistic, statistical, and logic-based methods [2, 7]. However, the recent emergence of large language models (LLMs) has opened new avenues for approaching this complex task. LLMs, with their ability to understand and generate human language, reason, and extract relevant information, offer the potential to streamline and automate aspects of ontology learning [3, 7].

Concurrently, the field of artificial intelligence has witnessed the rise of Agentic AI, a paradigm shift towards creating autonomous systems capable of pursuing complex goals with minimal human intervention [1]. Agentic AI systems demonstrate adaptability, advanced decision-making capabilities, and self-sufficiency, enabling them to operate dynamically in evolving environments [5]. This evolution in AI technology promises transformative applications across various domains, including ontology learning. However, the increasing agency of AI systems also raises important ethical considerations and challenges, particularly in terms of accountability and societal impact [5, 8].

One promising application of LLMs in the domain of ontology learning is direct ontology generation from text, often referred to as Text2OWL [7]. This approach aims to directly translate textual information into a formal ontology representation, typically using the Web Ontology Language (OWL). While this method holds significant promise, it can be challenging for longer and more complex texts, where a single LLM may struggle to maintain coherence and extract all relevant information.

To address these limitations, recent research has introduced the Chain-of-Agents (CoA) framework, a multi-agent system architecture that has demonstrated significant improvements in performance on long-context tasks [10]. CoA distributes the processing of input text across multiple "worker" agents, which collaboratively analyze different segments of the text and communicate their findings. A "manager" agent then synthesizes these contributions to produce a final, coherent output.

In this paper, we propose a novel approach that combines the power of LLMs with the Chain-of-Agents framework for ontology learning. By applying CoA to the LLM Text2OWL approach, we aim to overcome the challenges of coherence and completeness in automatically generated ontologies, particularly when dealing with extensive and intricate textual sources. We hypothesize that by breaking down the ontology generation task into smaller, more manageable steps and leveraging multiple LLM agents to collaborate on the solution, we can achieve more accurate and complete ontologies, especially when dealing with complex or lengthy textual sources.

The 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 ontology learning 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 ontology learning, contributing to the broader field of LLM-based multi-agent systems.

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 Ontology Learning approaches. In Section 3, we detail the proposed CoA-Text2OWL methodology, outlining the system architecture, agent prompts, and overall workflow. Section 4 presents the experimental setup, datasets, evaluation metrics, and the results obtained. Section 5 then offers a discussion of the findings, including performance comparisons, an analysis of the generated ontologies, and an examination of the impact of chain length and agent configurations, as well as limitations and challenges. Finally, Section 6 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: Agentic AI and Multi-Agent Systems, and Ontology Learning (Traditional and LLM-based approaches).

2.1 Agentic AI and Multi-Agent Systems

The field of Agentic AI is rapidly evolving, with a growing emphasis on creating autonomous systems capable of pursuing long-term goals, making decisions, and executing complex workflows [5]. Agentic AI shares conceptual roots with intelligent agents and multi-agent systems [9], but is distinguished by its capacity to tackle open-ended tasks and proactively generate novel solutions [5]. The survey by [4] provides an in-depth discussion of essential aspects of LLM-based multi-agent systems, including simulated environments, agent profiling, communication strategies, and mechanisms for capacity growth. LLM-based Multi-Agent systems offer advanced capabilities compared to systems using a single LLM-powered agent, because the LLMs can be specialized into various distinct agents, each with different capabilities, and enabling interactions among these diverse agents to simulate complex real-world environments effectively [4].

The Chain of Agents (CoA) approach [10] 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] (Mukherjee & Chang, 2025). 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 collaboratively perform ontology learning tasks, aiming to improve both 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].

2.2 Ontology Learning (Traditional and LLM-based approaches

Ontology learning is the process of automatically constructing ontologies from data [2]. Traditional approaches to ontology learning 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 ontology learning have emerged [7]. LLMs have been explored for various ontology learning tasks, including term typing [3], relationship extraction [3], and axiom generation [3, 7]. 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 ontology learning, comparing different LLM-based pipelines to traditional methods. Their study highlighted the potential of LLMs for automating aspects of ontology learning, but also identified limitations related to coherence and completeness. They evaluated three ontology learning 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 Chain-of-Agents framework [10], 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].

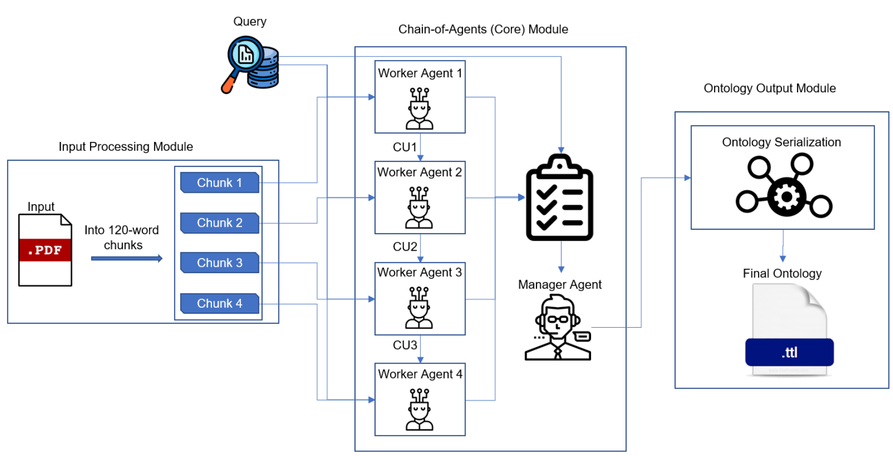

Fig. 1. CoA-Text2OWL Architecture

3 Methodology: CoA-Text2OWL

Our CoA-Text2OWL framework implements a novel approach to ontology generation from PDF documents (textual documents) using Chain-of-Agents (CoA) architecture [10]. This section details the system's components and workflow.

3.1 System Architecture

The CoA-Text2OWL system comprises three main components 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.

3.2 Agent Prompts

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 Chain-of-Agents architecture.

For worker agents, the prompt emphasizes their role in ontology construction and extracting key ontological elements from a specific text chunk:

> 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:

> 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 Chain-of-Agents approach.

3.3 Workflow

Input Processing 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.

Chain-of-Agents Processing The core of our system leverages a sequential Chain-of-Agents approach:

1. **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")
   - The output from the previous agent (initially None)

   Workers generate partial ontologies in Turtle format based on their assigned chunk.
2. **Manager Agent**: A single ManagerAgent, also using the open-mistral-7b model, synthesizes the outputs from all worker agents into a coherent final ontology.

Agent Communication 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.

Ontology Synthesis The manager agent performs the crucial task of integrating partial ontologies from worker agents. This process involves:

- Merging consistent concepts (classes) and relationships
- Resolving potential conflicts between partial ontologies
- Ensuring overall structural integrity of the final ontology

Output Generation The final ontology is serialized into a Turtle (.ttl) file, adhering to standard RDF and OWL conventions.

This methodology leverages 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 and Results

4.1 Datasets

Our experimental evaluation leveraged 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 ontology learning 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 applied—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 the first colonne (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 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 ontology learning 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 ontology learning 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.

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

where O_L represents the learned ontology and O_R represents the reference ontology.
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.

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

A high completeness score indicates that the ontology learning approach has successfully captured most of the domain-relevant concepts and relationships.
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.

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

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).

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 ontology learning 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 ontology learning.

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

4.5 Results

To evaluate the created ontologies, we begin with a quantitative analysis. Table 2 presents 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 analysis provides an overview of the structure and richness of the generated ontologies.

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

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).

Analysis of Results: 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 in this aspect.

Table 3. Comparison of ontology learning 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**

For class-level extraction, CoA-Text2OWL shows better precision by avoiding unnecessary elements, though its coverage of domain concepts remains only 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.
Overall, 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 ontology learning while pointing to areas for optimization.

5 Discussion

5.1 Performance Comparison

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 ontology learning, 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 ontology learning from text remains a challenging task, particularly for subclass extraction.

5.2 Limitations and Challenges

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 Concept 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 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 ontology learning, several challenges remain open for future research. An important direction involves refining the CoA framework to support ontology learning 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 ontology learning 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.

7 Acknowledgments

The authors thank the French company DAVI (Davi The Humanizers, Puteaux, France) for their support. This research 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).

References

1. Acharya, D.B., Kuppan, K., Divya, B.: Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey. IEEE Access (2025)
2. Armary, P., El-Vaigh, C.B., Narsis, O.L., Nicolle, C.: Ontology learning towards expressiveness: A survey. Computer Science Review **56**, 100,693 (2025)
3. Giglou, H.B., D'Souza, J., Sadruddin, S., Auer, S.: Llms4ol 2024 datasets: Toward ontology learning with large language models. In: Open Conference Proceedings, vol. 4, pp. 17–30 (2024)
4. Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O., Zhang, X.: Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680 (2024)
5. Mukherjee, A., Chang, H.H.: Agentic ai: Expanding the algorithmic frontier of creative problem solving. arXiv preprint arXiv:2502.00289 (2025)
6. Raad, J., Cruz, C.: A survey on ontology evaluation methods. In: International conference on knowledge engineering and ontology development, vol. 2, pp. 179–186. SciTePress (2015)
7. Schaeffer, M., Sesboüé, M., Charbonnier, L., Delestre, N., Kotowicz, J.P., Zanni-Merk, C.: 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 (2024)
8. Shavit, Y., Agarwal, S., Brundage, M., Adler, S., O'Keefe, C., Campbell, R., Lee, T., Mishkin, P., Eloundou, T., Hickey, A., et al.: Practices for governing agentic ai systems. Research Paper, OpenAI (2023)
9. Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. The knowledge engineering review **10**(2), 115–152 (1995)
10. Zhang, Y., Sun, R., Chen, Y., Pfister, T., Zhang, R., Arik, S.: Chain of agents: Large language models collaborating on long-context tasks. Advances in Neural Information Processing Systems **37**, 132,208–132,237 (2025)