Towards Adaptive Knowledge Structuring by Multi-Agent Consensus

Takahiro Kobayashi[1,*], Makoto Nakatsuji[1]

[1]NTT, Inc. Human Informatics Laboratories, 1-1, Hikari no Oka, Yokosuka city, Kanagawa, Japan

Abstract

As LLM-based agents become integral to workflows, enabling accurate task execution through collaboration between multiple agents and humans has emerged as a critical research challenge. Knowledge graphs (KGs) support semantic consistency but are often incomplete in dynamic domains. We propose a method combining (1) autonomous extraction of structured knowledge from inter-agent discussions, (2) integration into a shared evolving KG, and (3) consensus-driven refinement during task execution. Preliminary experiments compared two configurations: one implementing all components and another omitting consensus. Incorporating consensus increased non-taxonomic relations by 15% and improved relation precision from 79% to 97%, confirming its effectiveness. Future work will examine how refined knowledge impacts overall task accuracy.

Keywords

Knowlede Graph, Large Language Models, LLM-based Agents, Human-AI Collaboration, Multi-Agent Systems

1. Introduction

As AI agents powered by large language models (LLMs) become increasingly integrated into real-world workflows, there is growing demand for accurate and reliable task execution. Collaborative frameworks involving multiple agents [1, 2, 3] and human-in-the-loop approaches [4] have been proposed to align task outputs with human intent. While many tasks require up-to-date domain knowledge, LLMs only store knowledge learned during pretraining. Fine-tuning is one way to incorporate new knowledge, but it demands enormous time and computational resources, making it impractical for frequent updates.

Retrieval-Augmented Generation (RAG) [5] addresses this by combining LLMs with external knowledge bases. Among candidates, vector-search-based document databases [6] and knowledge graphs (KGs) [7] are promising. Vector-based stores retrieve semantically similar documents but cannot guarantee accurate relationships. In contrast, KGs explicitly represent semantic relations, making them suitable for complex reasoning and consistency checks.

To utilize KGs in RAG, it is necessary to construct the graph itself. Public KGs such as Wikidata[1] are often insufficient for domains with fluid structures, where knowledge must adapt to emerging entities and shifting relationships. Agents must therefore update and reorganize knowledge representations to respond effectively to changing requirements.

This study proposes a framework to enhance task accuracy by organizing and sharing knowledge extracted from input materials. It consists of three components: (1) autonomous extraction of structured knowledge; (2) construction of a shared KG; and (3) consensus-based refinement through "KG update meetings." To validate the approach, we conducted experiments using ACT-generated dialogue logs [1] as input. Results show that consensus-based refinement increased the qunantity of non-taxonomic relations by 15% and improved precision from 79% to 97%.

HAIBRIDGE'25: the 1st Workshop on Bridging Hybrid (Artificial) Intelligence and the Semantic Web November 2-3, 2025, Nara, Japan

*Corresponding author.

tkhr.kobayashi@ntt.com (T. Kobayashi); tkhr.kobayashi@ntt.com (M. Nakatsuji)

[1]https://www.wikidata.org/

2. Preliminaries

2.1. Teminology

To ensure clarity, we define key terms used throughout this paper:
Task Resolution. The process of generating an output artifact that satisfies a user-specified task instruction. Task resolution involves interpreting the input task and producing a coherent deliverable aligned with the given requirements.
Agent. In this study, an *agent* refers to an AI entity that performs a specific role based on a consistent objective and persona. Each agent operates by selectively employing multiple LLM prompts to accomplish its assigned responsibilities within the collaborative workflow.
Input Materials. We use the term *input materials* to denote the data targeted for knowledge extraction during KG updates. Typically, these materials consist of textual resources—structured or unstructured—obtained through autonomous research activities such as web searches, database queries, or interactions with external expert agents.

2.2. LLMs4OL

The LLMs4OL framework [8] investigates the hypothesis that LLMs known for their ability to capture complex linguistic patterns, can effectively support ontology learning (OL) tasks without extensive domain-specific training. Specifically, LLMs4OL evaluates multiple LLM families under zero-shot prompting for three core OL tasks:

- **Term Typing**: Assigning ontological types to extracted terms.
- **Taxonomy Discovery**: Identifying hierarchical relationships among concepts.
- **Non-Taxonomic Relation Extraction**: Detecting semantic relations beyond hierarchical structures.

Empirical results show that LLMs generalize effectively across diverse ontological domains, including lexical (WordNet), geographical (GeoNames), and biomedical (UMLS), offering a scalable alternative to manual curation.

Our research builds upon this paradigm by adopting the same three tasks as foundational components of our methodology to build KGs. By leveraging LLMs in these processes, we aim to enhance semantic consistency, aligning with the principles established by LLMs4OL.

2.3. ACT

Our research assumes a framework in which multiple agents autonomously conduct researches and collaboratively generate outputs, similar in spirit to ACT [1]. ACT operationalizes complex task execution through a four-phase workflow:

1. **Agent Generation**: The input task is decomposed into subtasks, and agents are dynamically instantiated and assigned responsibility for each subtask.
2. **Team Meeting**: Agents engage in a collaborative discussion to refine the overall task design and their respective subtasks. A randomly selected leader *fuses* individual task proposals into a coherent team-level plan, ensuring feasibility and alignment; agents also acquire structured knowledge of others' approaches.
3. **Break Time**: Each agent independently analyzes its contribution and deepens its expertise. To that end, agents dynamically generate expert agents and conduct focused interviews to expand task-relevant knowledge.
4. **Production Meeting**: Agents share accumulated insights and integrate their contributions to produce the final output aligned with the team-level task.

Inspired by this paradigm, our study proposes a method for consolidating the outcomes of agents' discussions and the findings from their autonomous research into a formal KG. Rather than limiting knowledge to ACT's internal representation for coordination, we leverage agents' interactions and external knowledge sources to populate and refine a KG.

3. Methodology

3.1. Overview of Task Resolution

Let x denote an input task specification and y the desired final output. The framework resolves x through four phases, while maintaining a persistent knowledge graph (KG) that evolves across tasks:

1. **Agent Generation.** The task x is decomposed into subtasks $\mathcal{S} = \{s_1, \ldots, s_m\}$. For each s_j, we instantiate an agent a_i with a persona π_i and a role-specific task descriptor ϑ_i, forming a team $A = \{a_1, \ldots, a_n\}$.
2. **Individual Research.** Each agent autonomously gathers input materials M_i (e.g., web/DB documents or outputs from external agents) guided by (π_i, ϑ_i) and a KG-conditioned view V_i of the current graph $KG^{(t)}$. Formally,

$$V_i \;=\; \mathrm{Select}\big(KG^{(t)},\, \pi_i,\, \vartheta_i\big), \qquad M_i \;=\; \mathrm{Research}\big(\pi_i,\, \vartheta_i,\, V_i\big).$$

3. **KG Update Meeting.** Agents propose knowledge extractions from $\{M_i\}$ under a consensus protocol (Sec. 3.3), yielding a consolidated update ΔKG and an updated graph $KG^{(t+1)}$.
4. **Output Production Meeting.** Referencing $KG^{(t+1)}$, agents iteratively co-author the final output y under the same consensus protocol.

The KG is reused across tasks and updated monotonically with conflict handling, enabling long-horizon knowledge reuse and consistent decision-making. This study focuses on KG generation and does not specify details of individual research or output production beyond aspects relevant to KG construction.

3.2. Knowledge Extraction Pipeline

The knowledge graph at iteration t is represented as

$$KG^{(t)} = \big(T^{(t)},\, C^{(t)},\, R^{(t)}_{\mathrm{NT}},\, R^{(t)}_{\mathrm{TAX}}\big),$$

where:

nosep $T^{(t)}$: set of terms (entities/concepts),
nosep $C^{(t)}$: set of types (the first level classes of the taxonomy),
nosep $R^{(t)}_{\mathrm{NT}}$: set of non-taxonomic relations,
nosep $R^{(t)}_{\mathrm{TAX}}$: set of taxonomic (IS-A) relations.

Given input materials M collected by agents, the update process for KG consists of the following steps:

(i) Term Extraction. Extract domain-specific terms from M:

$$T^{\mathrm{cand}} = g_{\mathrm{term}}(M)$$

Merge with existing terms:

$$T^{(t+1)} = T^{(t)} \cup T^{\mathrm{cand}}.$$

(ii) Non-Taxonomic Relation Extraction. Identify semantic relations among terms:

$$R_{\text{NT}}^{\text{cand}} = g_{\text{nt}}(T^{(t+1)}, M).$$

Merge with existing relations:

$$R_{\text{NT}}^{(t+1)} = R_{\text{NT}}^{(t)} \cup R_{\text{NT}}^{\text{cand}}.$$

(iii) Type Extraction. Generate candidate types incrementally:

$$C^{(t+1)} = g_{\text{type}}\big(T^{(t+1)}, C^{(t)}, M\big).$$

This design allows dynamic type induction without relying on a fixed schema.

(iv) Type Assignment. Assign each term in $T^{(t+1)}$ to exactly one type in $C^{(t+1)}$. Formally:

$$\alpha : T^{(t+1)} \to C^{(t+1)},$$

where $\alpha(\tau)$ returns the unique type $c_j \in C^{(t+1)}$ that best represents term τ. If no suitable type exists, a new type candidate may be introduced during g_{type} in the next iteration.

(v) Taxonomic Relation Extraction. For each type $c_j \in C^{(t+1)}$, extract hierarchical relations using:

$$R_{\text{TAX},j} = g_{\text{tax}}\big(c_j,\ \{\tau \in T^{(t+1)} \mid \alpha(\tau) = c_j\}\big).$$

Aggregate all per-type results:

$$R_{\text{TAX}}^{(t+1)} = \bigcup_{c_j \in C^{(t+1)}} R_{\text{TAX},j}.$$

The updated knowledge graph is:

$$KG^{(t+1)} = \big(T^{(t+1)}, C^{(t+1)}, R_{\text{NT}}^{(t+1)}, R_{\text{TAX}}^{(t+1)}\big)$$

Conflict resolution and structural invariants (e.g., acyclicity for R_{TAX}) are enforced during merge.

3.3. Multi-Agent Consensus Algorithm

The consensus mechanism operates as an internal control loop within each stage of the knowledge extraction pipeline described in Section 3.2. Specifically, whenever a pipeline step—such as term extraction, type induction, or relation identification—is executed, multiple agents collaborate to refine the intermediate outputs before committing them to the evolving knowledge graph. **Agent state.** Each agent a_i is parameterized by $s_i = (\pi_i, \vartheta_i)$, where π_i encodes its persona and ϑ_i specifies the role-dependent extraction task. The task descriptor determines which pipeline function the agent invokes, e.g., Extract(s_i, M_i), where Extract corresponds to one of the operators $g_{\text{term}}, g_{\text{type}}, g_{\text{nt}}, g_{\text{tax}}$ defined in Section 3.2.

Iterative consensus protocol. For a given pipeline step, the process unfolds as follows:

1. **Initial Proposal.** Each agent a_i produces an initial candidate $\Delta_i^{(0)} = \text{Extract}(s_i, M_i)$, applying its designated extraction operator to its local input materials.
2. **Opinion Statement.** At iteration k, agent a_i emits an opinion $o_i^{(k)} = (\beta_i^{(k)}, \delta_i^{(k)}, \rho_i^{(k)})$, where $\beta_i^{(k)}$ denotes elements to *modify* in the current shared candidate set, $\delta_i^{(k)}$ denotes elements to *add* and $\rho_i^{(k)} \in \text{True}, \text{False}$ denotes agreement flag indicating whether the agent accepts the current state.
3. **Opinion Aggregation.** A designated leader ℓ_k integrates all opinions into a consolidated agenda $\overline{\Omega}^{(k)} = \text{Integrate}(o_i^{(k)}{}_{i=1}^{n})$, summarizing required modifications and additions.

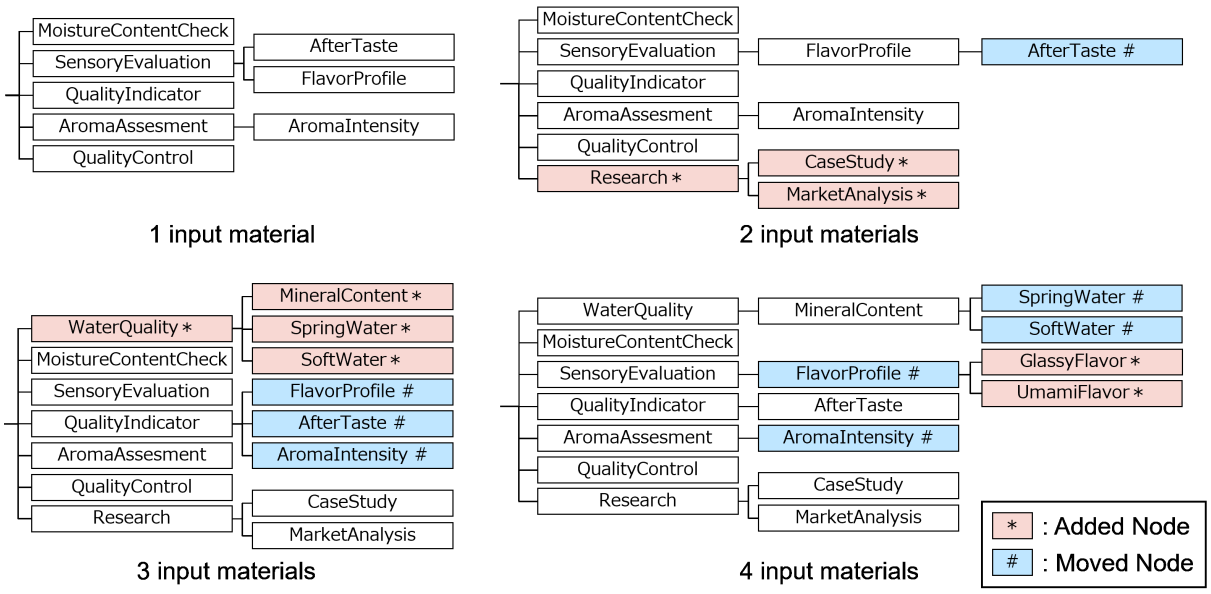

Figure 1: Evolution of the hierarchical structure within the "QualityAssessment" type. Newly added nodes are indicated with asterisks(∗), and nodes with relationships changed are indicated with hashes(#).

4. **Regeneration.** Each agent revises its proposal conditioned on $\overline{\Omega}^{(k)}$: $\Delta_i^{(k+1)} = \text{Regenerate}(s_i, \overline{\Omega}^{(k)})$.

The loop terminates when all agents agree ($\forall i : \rho_i^{(k)} = \text{True}$) or the maximum iteration count $T_{\max}$ is reached (three in our experiments). The finalized proposals are then merged into the global knowledge graph using a conflict-aware operator. This design ensures that consensus-driven refinement occurs at every pipeline stage, yielding a more coherent and structurally consistent $KG^{(t+1)}$.

4. Experiments

4.1. Datasets and settings

We simulated a scenario where input materials arise from agents conducting individual research and interacting with external agents. For this simulation, we employed the ACT framework to generate input materials. ACT was executed using the Reddit Creative Writing dataset [1], which consists of 6,673 non-factoid QA pairs extracted from Reddit posts and categorized into five domains: Tea, Cafe, Design, Architecture, and Fashion. From the Tea category, we randomly selected four questions and assigned each as a subtask to one of four agents. The agents then performed the Team Meeting and Break Time (cf. section 2.3) phases of ACT. During Break Time, each agent instantiated a domain-specific expert agent and conducted an interview of 24–26 turns, resulting in four dialogue logs. These logs were treated as input materials for our framework (denoted as M_1 through M_4 in Section 3.1). We conducted two experiments:

Experiment 1: We varied the number of input materials $|M|$ from 1 to 4 to examine how KG evolves as more materials are incorporated. In this setting, consensus-based refinement was omitted; initial proposals $\Delta^{(0)}$ were directly applied to KG updates. Experiment 2: Using all four input materials, we compared KG construction with and without consensus, focusing on the number and precision of non-taxonomic relations. Precision was computed as $TP/(TP + FP)$. The authors manually inspected each input material: a relation was labeled true positive if its semantics were supported by the material, and false positive otherwise.

All experimental processes were implemented using GPT-4o mini.

4.2. Experiment 1: Progressive KG Development

This experiment is aimed to assess the feasibility of incremental knowledge graph (KG) expansion. We constructed four KGs using one to four input ma- terials to analyze graph evolution. As input increased, the KG is expanded to more detailed structures with non-taxonomic relationships. Ultimately, nine types were extracted: BrewingFactors, HealthImpact, OdorManagement, Qual- ityAssessment, SellerCategory, StorageMethod, TeaOrigin, TeaStoragePractices, and TeaType.

Findings from Structural Evolution. Figure 1 presents the evolution of the hierarchical structure with respect to the `QualityAssessment` type as input materials are incrementally incorporated. Our analysis yields the following three findings: (1) Despite being independently generated, the four graphs exhibit recurring structural patterns. This indicates a degree of logical reproducibility in how agents organize knowledge, suggesting that the process is not entirely arbitrary. (2) In contrast, nodes such as `FlavorProfile` show high positional variability across graphs. This suggests possible inconsistencies in terminology or semantic boundaries, highlighting areas where the schema may require re- finement. (3) These observations underscore the importance to evaluate the ro- bustness of the knowledge structure when incorporating new information. Such evaluation serves as a diagnostic tool to identify unstable areas and assess the overall coherence and validity of the resulting graph.

4.3. Experiment 2: Refining the KG via Multi-Agent Consensus

We compared KG construction with and without the consensus mechanism. The number of the relations increased from 33 to 38, and the precision improved from 79% to 97%. Agents' consensus corrected errors such as reversed entities (e.g., `('BakingSoda', 'uses', 'OdorNeutralization')`). Challenges included concretizing general relations and simplifying overly descriptive types. For instance, `(Darjeeling, hasBrewingTechnique, BrewingTechnique)` lacked specific attributes, and `(Tea, benefitsFromStorageTechnique, CoolAndDryPlace)` could be better structured. Such improvements are a subject for future work.

5. Conclusion

This study proposed a framework in which multiple AI agents collaboratively execute tasks while constructing and refining a knowledge graphs (KGs) in do- mains with fluid knowledge structures. Using an LLM-based knowledge extrac- tion, we built a KG incorporating both taxonomic and non-taxonomic relations and analyzed its evolution quality through repeated task execution. Our evaluation implies that the KG structure exhibits varying stability— some components remain consistent across tasks, while others require refinement. This suggests a strategy for improving reliability by focusing on unstable regions. We also demonstrated that multi-agent consensus enhances the quality of extracted knowledge. Specifically, the proportion of logically valid non-taxonomic

relationships was improved from 79% to 97%, confirming the effectiveness of collaborative refinement. Future work will explore algorithmic improvements to further enhance the accuracy in creating KGs and in resolving tasks using created KGs.

Declaration on Generative AI

During the preparation of this work, the authors used Microsoft 365 Copilot in order to: drafting content, grammar and spelling check and content enhancement. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. Nakatsuji, S. Tateishi, Y. Fujiwara, A. Matsumoto, N. Nomoto, Y. Sato, ACT: Knowledgeable agents to design and perform complex tasks, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar

(Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2025, pp. 16831–16861.

[2] I. Abbasnejad, X. Liu, A. Roy, Deciding the path: Leveraging multi-agent systems for solving complex tasks, in: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR) Workshops, 2025, pp. 4216–4225.

[3] R. Barbarroxa, L. Gomes, Z. Vale, Benchmarking large language models for multi-agent systems: A comparative analysis of autogen, crewai, and taskweaver, in: P. Mathieu, F. De la Prieta (Eds.), Advances in Practical Applications of Agents, Multi-Agent Systems, and Digital Twins: The PAAMS Collection, Springer Nature Switzerland, 2025, pp. 39–48.

[4] W. Takerngsaksiri, J. Pasuksmit, P. Thongtanunam, C. Tantithamthavorn, R. Zhang, F. Jiang, J. Li, E. Cook, K. Chen, M. Wu, Human-in-the-loop software development agents, in: Proceedings of the 47th IEEE/ACM International Conference on Software Engineering, ICSE 2025, ACM, 2025.

[5] T. T. Procko, O. Ochoa, Graph retrieval-augmented generation for large language models: A survey, in: 2024 Conference on AI, Science, Engineering, and Technology (AIxSET), 2024, pp. 166–169.

[6] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with gpus, IEEE Transactions on Big Data 7 (2021) 535–547.

[7] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Comput. Surv. 54 (2021).

[8] H. Babaei Giglou, J. D'Souza, S. Auer, 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, Springer Nature Switzerland, 2023, pp. 408–427.