

A Collaborative Knowledge Graph Development Agent for Neurosymbolic AI Agents

Spencer M. Seals^{1,*},†

¹Wright State University, 3640 Colonel Glenn Hwy. Dayton, Ohio, USA 45435-0001

Abstract

Knowledge graphs have gained much attention as a means of controlling the behavior and outputs of Artificial Intelligence agents. However, traditional approaches to building knowledge graphs require a considerable amount of effort. These projects require knowledge graph developers trained in computer science and formal logic and domain experts who can identify relevant concerns and provide feedback. To address this challenge, I have developed a collaborative knowledge graph development agent. This agent allows users to provide a target application area. The agent then steps through creating a knowledge graph following a knowledge graph engineering method. At each step, the user can pose questions and provide feedback to shape the agent's behavior. The agent then produces a knowledge graph and associated documentation. This collaborative agent makes knowledge graph development more accessible to people who lack extensive experience with knowledge graph development. This allows domain experts to engage in an iterative development process and produce a final knowledge graph that can be used to control AI agents with minimal training required.

Keywords

large language models, knowledge graph, knowledge graph creation

1. Introduction

The growing popularity and accessibility of large language models (LLMs) has prompted increased interest in using LLMs to assist with the development and expansion of knowledge graphs. This stream of research proposes that because LLMs are trained on large corpora that cover variety of topics, they encode useful information in their parameter weights that may be used for aiding in the construction and completion of knowledge graphs.

A number of papers have proposed LLM powered approaches to performing various knowledge engineering tasks ranging from assisting users with using knowledge graphs to creating entire knowledge graphs from scratch [1, 2, 3, 4, 5, 6, 7]. However, all of these approaches are focused on developing fully automated methods. If these methods involve any human intervention, this intervention primarily occurs in the context of evaluating the final output. This work assumes LLMs have the ability to generate effective knowledge graphs in isolation. However, for most practical knowledge engineering projects, this may not be a reasonable assumption (e.g., due to technical domain content in the target domain). Moreover, human feedback during the development process can improve the final knowledge graph and prevent errors and incorrect conceptualizations that may be introduced by LLMs [8].

In this work, I present a collaborative knowledge graph development agent that is currently under development. This agent allows a human user to provide a target topic for a knowledge graph and assists the user with the initial development process of creating a knowledge graph. This approach has the added benefit of allowing domain experts who may have limited or no experience developing knowledge graphs to develop and refine an initial prototype. This work follows previous work that has proposed methods to assist domain experts with editing existing knowledge graphs [9].

K-CAP 2025 Companion Volume

*The views expressed are those of the authors and do not necessarily reflect the official policy or position of the Department of the Air Force, the Department of Defense, or the U.S. government. Approved for public release, case number: AFRL-2026-0650

**Work supported by DAGSI RH-10-24.

✉ s.m.seals@outlook.com (S. M. Seals)

ORCID 0000-0002-3175-2681 (S. M. Seals)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

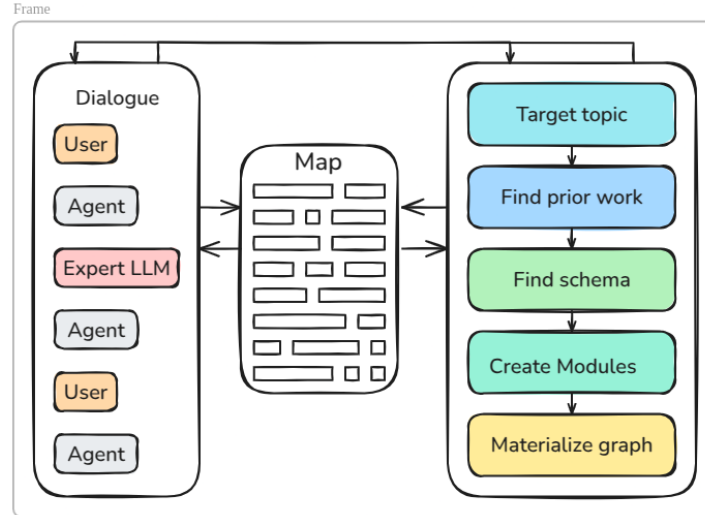


Figure 1: An overview of the agent. The dialogue column on the left represents interactions between the end user, the discourse management LLM, and any (optional) expert LLMs that the user has decided to include. Results from these interactions are added to a map structure. The right hand column contains different tasks in the modular ontology modeling framework [10]. These tasks are performed through combining results from the user dialogue with the agent and results generated from the different LLMs that perform tasks in the background. These LLMs include the question asking and answering LLMs, knowledge base LLM, outline generation and outline refinement LLMs, and the concept extraction LLM. These LLMs also contribute to and retrieve data from the map structure. The materialize graph box in the lower right hand corner represents future work.

2. Methods

In this section, I briefly describe the methods used to develop the agent. An overview of the agent framework is located in Figure 1.

2.1. Design Approach

The proposed knowledge graph development agent is designed to assist a user that has minimal experience with knowledge graphs develop an initial knowledge graph that adequately represents the target domain. To maximize the functionality of the knowledge graph that the agent produces, I opted to use a formal knowledge engineering approach. This helps ensure that the knowledge graph that the agent produces can be successfully used in downstream applications. Moreover, I expect that knowledge graphs developed using this tool may need to be edited after creation either to correct problems with the original graph or to add additional information or concepts to the graph. With these requirements in mind, I opted to have this agent use a modular ontology method to develop knowledge graphs [10]. Modular ontology modeling uses a pattern based method that focuses on modeling a set of key constructs and their relations. This pattern based approach makes it easy to update an ontology for a new related use case, integrate the existing ontology with other sources of data, or update the ontology when new information becomes available.

Currently, the agent allows a human user to provide a target topic for a knowledge graph and assists the user with refining their target use case, developing competency questions, preparing a use case report, and identifying modules that represent key concepts in the target domain. Throughout, the user can provide feedback to refine and guide the generated knowledge graph.

2.2. Agent Approach

The agent uses multiple interacting LLMs to perform different parts of the knowledge graph development process. This approach follows other work that uses interacting LLMs to generate simulated dialogue

from multiple perspectives and develop interactive agents [11, 12, 13]. The agent uses seven different LLMs to perform tasks in the process. LLMs can be initialized as different instances of the same LLM or can be customized for specific tasks. Supported LLMs include: Open AI, Deep Seek, Azure, Groq, Google, Anthropic, vLLM server, TGIClient, and Ollama. As part of the development process, the agent retrieves relevant information either at run time via the web or via information retrieved from a local vector database. Supported options include: you.com, Bing, Google, serper.dev, Brave, Sear XNG, Duck Duck Go, Tavily, Azure AI, and Qdrant vector databases.

The seven tasks are discourse management, question asking, question answering, knowledge base, outline generation, outline refinement, and concept extraction. The end user of the system primarily interacts with the discourse management LLM. This LLM asks the user questions and provides feedback on what the overall agent is doing to the user. This LLM also coordinates the behavior of the LLMs that perform the other tasks. The question asking and question answering LLMs interact with the discourse management LLM to conduct a simulated dialogue about the target domain for the knowledge graph proposed by the user. These LLMs ask and answer clarifying questions and identify information that needs to be retrieved to successfully develop the knowledge graph. The knowledge base LLM assists with retrieving and organizing information identified from the simulated discussion. This LLM either retrieves information from the internet or from a local vector database. The outline generation and refinement LLMs develop and refine the use case report using feedback from the user, the interactions between the other LLMs, and knowledge retrieved from the knowledge base LLM. The concept extraction LLM extracts key concepts and develops modules using the outputs of the other LLMs and user feedback. In the current iteration of the agent, the concept extraction piece relies on a combination of general reasoning abilities of the LLM selected for the concept extraction task and feedback from the user about the overall quality and specificity of the concepts. Individual LLMs interact with the outputs of the other LLMs through the use of a map structure. This allows each LLM to generate responses and actions that are consistent with what other LLMs that perform different tasks have generated.

3. Conclusion

This work presents a collaborative approach to LLM assisted knowledge graph development. This approach allows human users to interact with an agent to develop and refine an initial prototype knowledge graph. This does not require extensive expertise on the design and development of knowledge graphs and automates a large amount of manual effort traditionally required to construct a knowledge graph.

There are several opportunities for future research related to this project. First, is evaluating the agent with real users and domain experts. Additionally, there are a number of opportunities for refining the agent's capabilities. Currently, many of the agents capabilities rest on a combination of inherent abilities of LLMs combined with feedback from human users. Particularly for some tasks like concept extraction, this may influence the overall quality of the results and will be addressed as part of the user evaluation. Lastly, while the use of a formal knowledge engineering methodology does provide structural consistency, it does not guarantee the logical consistency of the final graph. The current iteration of the agent relies on feedback and input from the user to review and verify logical consistency of the final knowledge graph. In future work, we plan to explore automated and semi-automated ways of addressing this concern.

Declaration on Generative AI

The author did not employ any Generative AI tools during the preparation of this manuscript and takes full responsibility for the publication's content.

References

- [1] Y. Rebboud, L. Tailhardat, P. Lisena, R. Troncy, Can LLMs Generate Competency Questions?, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A. G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), *The Semantic Web: ESWC 2024 Satellite Events*, volume 15344, Springer Nature Switzerland, Cham, 2025, pp. 71–80. URL: https://link.springer.com/10.1007/978-3-031-78952-6_7, series Title: Lecture Notes in Computer Science. doi:10.1007/978-3-031-78952-6_7.
- [2] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An LLM supported approach to ontology and knowledge graph construction, 2024. URL: <http://arxiv.org/abs/2403.08345>. doi:10.48550/arXiv.2403.08345, arXiv:2403.08345 [cs].
- [3] V. K. Kommineni, B. König-Ries, S. Samuel, Towards the automation of knowledge graph construction using large language models, in: *Proceedings of the NLP4KGC: 3rd international workshop on natural language processing for knowledge graph creation*, Amsterdam, Netherlands, 2024. URL: <https://ceur-ws.org/Vol-3874/paper2.pdf>.
- [4] M. Zhang, X. Ye, Q. Liu, P. Ren, S. Wu, Z. Chen, Knowledge Graph Enhanced Large Language Model Editing, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 22647–22662. URL: <https://aclanthology.org/2024.emnlp-main.1261/>. doi:10.18653/v1/2024.emnlp-main.1261.
- [5] H. Babaei Giglou, J. D’Souza, F. Engel, S. Auer, LLMs4OM: Matching Ontologies with Large Language Models, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A. G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), *The Semantic Web: ESWC 2024 Satellite Events*, volume 15344, Springer Nature Switzerland, Cham, 2025, pp. 25–35. URL: https://link.springer.com/10.1007/978-3-031-78952-6_3, series Title: Lecture Notes in Computer Science. doi:10.1007/978-3-031-78952-6_3.
- [6] K. Karakeben, H. E. Sorotos, A. Milchevskaya, A. Assaf, Rapid Graph Generation from Job Descriptions: Combining Taxonomies and LLMs, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A. G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), *The Semantic Web: ESWC 2024 Satellite Events*, volume 15344, Springer Nature Switzerland, Cham, 2025, pp. 143–146. URL: https://link.springer.com/10.1007/978-3-031-78952-6_15. doi:10.1007/978-3-031-78952-6_15, series Title: Lecture Notes in Computer Science.
- [7] F. Zhang, Y. Ding, J. Cheng, ATAP: Automatic Template-Augmented Commonsense Knowledge Graph Completion via Pre-Trained Language Models, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 16456–16472. URL: <https://aclanthology.org/2024.emnlp-main.919/>. doi:10.18653/v1/2024.emnlp-main.919.
- [8] L.-P. Meyer, C. Stadler, J. Frey, N. Radtke, K. Junghanns, R. Meissner, G. Dziwis, K. Bulert, M. Martin, LLM-assisted Knowledge Graph Engineering: Experiments with ChatGPT, in: C. Zinke-Wehlmann, J. Friedrich (Eds.), *First Working Conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow*, Springer Fachmedien Wiesbaden, Wiesbaden, 2024, pp. 103–115. URL: https://link.springer.com/10.1007/978-3-658-43705-3_8. doi:10.1007/978-3-658-43705-3_8, series Title: Informatik aktuell.
- [9] I. Celino, V. A. Carriero, A. Azzini, I. Baroni, M. Scrocca, Procedural knowledge management in Industry 5.0: Challenges and opportunities for knowledge graphs, *Journal of Web Semantics* 84 (2025) 100850. URL: <https://www.sciencedirect.com/science/article/pii/S1570826824000362>. doi:10.1016/j.websem.2024.100850.
- [10] C. Shimizu, K. Hammar, P. Hitzler, Modular ontology modeling, *Semantic Web* 14 (2023) 459–489. doi:10.3233/SW-222886.
- [11] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, C. Wang, AutoGen: Enabling next-gen LLM applications

via multi-agent conversations, in: First conference on language modeling, 2024. URL: <https://openreview.net/forum?id=BAakY1hNKS>.

- [12] Y.-M. Tseng, Y.-C. Huang, T.-Y. Hsiao, W.-L. Chen, C.-W. Huang, Y. Meng, Y.-N. Chen, Two Tales of Persona in LLMs: A Survey of Role-Playing and Personalization, in: Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 16612–16631. URL: <https://aclanthology.org/2024.findings-emnlp.969>. doi:10.18653/v1/2024.findings-emnlp.969.
- [13] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, Z. Liu, ChatEval: Towards better LLM-based evaluators through multi-agent debate, in: The twelfth international conference on learning representations, 2024. URL: <https://openreview.net/forum?id=FQepisCUWu>.