Building Hierarchy-Aware Knowledge Graphs: Ontology-Grounded Triple Extraction with LLMs*

Kudzai Sauka[1,2,*,†], Gianluigi Bardelloni[3,†], Jigsa Bulto[3] and Frederik B. I. Situmeang[1,2]

[1] *Amsterdam University of Applied Sciences, Fraijlemaborg 133, 1102 CV, Amsterdam, The Netherlands*
[2] *University of Amsterdam, Plantage Muidergracht 12, 1018TV, Amsterdam, The Netherlands*
[3] *KPN, Teleportboulevard 121, 1043 EJ, Amsterdam, The Netherlands*

Abstract

This paper introduces a hierarchy-aware framework, Document-Preprocessing-Extract-Resolve-Merge-Canonicalize (DERMC), for constructing knowledge graphs using large language models (LLMs). It addresses the limitations of naive LLM prompting in creating a knowledge graph, which often results in redundancy, LLM cognitive overload, and inconsistencies across languages. Building on the Extract-Define-Canonicalize (EDC) paradigm, our approach integrates multilingual coreference resolution, hierarchical document parsing, and a RAG-MCP-inspired schema retriever that dynamically narrows candidate relations for each document context, thereby reducing the size of the LLM prompt. The system supports both commercial and local LLM backends, allowing flexible deployment. Preliminary results show improved alignment between flat and hierarchical parsing modes. Full-scale experiments and evaluations, including assessments by industry and knowledge representation experts, are planned to validate performance and quality in enterprise contexts

Keywords

Knowledge Graph, Coreference Resolution, Schema Alignment, Triple Extraction, Ontology

1. Introduction

The demand for non-hallucinating, transparent Gen-AI chatbots continues to grow as businesses aim to address user trust and consumer acceptance [1, 2, 3]. Using retrieval-augmented generation (RAG) with knowledge graphs (KGs) in Gen-AI chatbots helps reduce hallucinations and improves transparency [4, 5, 2]. However, creating KGs from unstructured text remains challenging, particularly in customer service, where chatbots ground answers in knowledge articles. KG construction is typically a manual and cumbersome process that requires domain experts and knowledge representation specialists [6, 7, 8, 9]. This manual process is time-consuming, costly, and challenging to scale for large, rapidly changing domains [7, 10]. Current methods to speed up KG building involve using the language understanding capabilities of large language models (LLMs) [11, 8, 6]. However, applying LLMs directly to KG creation has drawbacks, such as generating inconsistent or duplicate triples due to the absence of an explicit, unified schema [7, 8, 9].

2. Industry Challenges

In industry use cases, the drawbacks of directly applying LLMs to auto-create KGs are worsened by the hierarchical structure of the documents and large ontologies. Knowledge documents are organized with headings to facilitate navigation and present information in a hierarchical manner. Without awareness of this hierarchy, context can be lost, and triples may miss the correct parent-child relationships. Another challenge is that the generated KGs may be incomplete or biased toward the LLM's training

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan
*Corresponding author.
†These authors contributed equally.
k.sauka@hva.nl (K. Sauka); gianluigi.bardelloni@kpn.com (G. Bardelloni); jigsa.bulto@kpn.com (J. Bulto); f.b.i.situmeang@hva.nl (F. B. I. Situmeang)
https://ksauka.github.io/ (K. Sauka)
0000-0002-3233-895X (K. Sauka); 0009-0008-6134-1972 (G. Bardelloni); 0000-0002-2156-2083 (F. B. I. Situmeang)

data, which may not fully cover the target domain, especially for proprietary documents not included in the pre-training dataset. Additionally, if domain ontologies are available, prompts for LLMs become very large when integrating knowledge articles with entire domain ontologies, thereby increasing costs and slowing inference. Furthermore, our industry's specific use cases, knowledge documents, and ontologies often involve mixed languages.

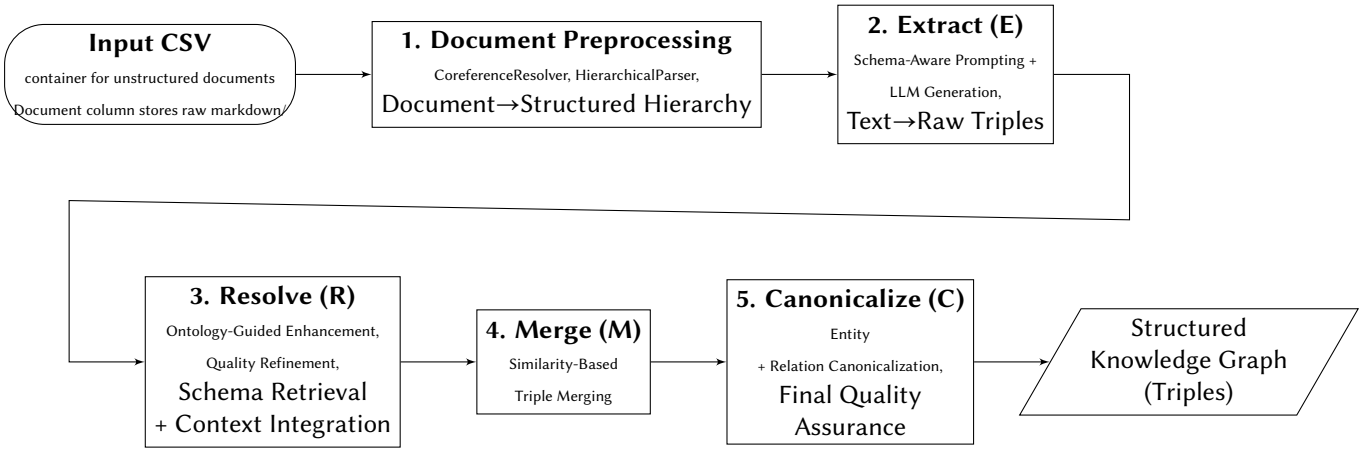

Figure 1: DERMC framework. figure design credit[12]

3. Proposal

To address challenges from naive LLM prompting in knowledge graph construction, we propose DERMC, a modular framework extending Zhang and Soh's [6] Extract–Define–Canonicalize plus Refinement (EDC+R) paradigm. Our system incorporates hierarchy awareness, multilingual coreference resolution, and similarity-based triple merging with redundancy reduction (see Figure 1). Without section awareness, relations can leak across scopes, causing misinterpretations and duplication. Our experiment with a simulated dataset confirms the benefits of this approach [1].

Unlike Zhang and Soh's [6] six-prompt approach, our pipeline reduces LLM reliance by replacing the original "Define" phase with schema-context-based resolution. Consequently, DERMC requires just two LLM prompts: one for extraction (incorporating ontology relations and document hierarchy) and another for refinement (using retrieved ontology candidates to enhance triple quality). All other components—coreference resolution, hierarchical parsing, similarity calculations, and canonicalization—utilize specialized models and algorithms, which we anticipate will significantly reduce computational overhead and LLM costs.

Following [6], our schema-driven resolution precomputes dense embeddings of schema relations and entity types. To improve beyond label-only matching, we encode ontology predicates with relation labels, synonyms, glosses, class labels, predicate phrases, and sentence windows, then compute cosine similarity. The method falls back on label and class labels when data is missing to maintain accuracy and compactness. When scores are close, the top candidate is selected with an ambiguity flag, balancing failure visibility and recall. These embeddings enable dynamic retrieval of only the top-k relevant candidates per document segment, reducing token requirements, operational costs, and hallucination risks. This retrieval step draws inspiration from the RAG-MCP architectures in [13], which aim to narrow candidates for efficient context window management. Our implementation differs from the EDC+R approach in [6], which utilized a fine-tuned E5-Mistral-7b-Inst embedding model trained with the InfoNCE loss on the TekGen dataset. Instead, we employ the off-the-shelf all-MiniLM-L6-v2 model, which provides strong semantic understanding for short texts like relations and entities.

[1]We have provided an example implementation code and instructions to generate the synthetic data in the README file./ https://github.com/ksauka/DERMC-.git

4. Framework

The DERMC framework 1 begins with document preprocessing, where we incorporate a multilingual coreference resolver based on FastCoref's LingMess architecture, combined with translation pipelines like Facebook's M2M100, enabling smooth handling of mixed-language documents, including those with both Dutch and English. Additionally, we developed a custom DocumentParser to process markdown-structured documents, preserving their hierarchical structure, which is essential for maintaining contextual relationships during downstream triple extraction.

During the extraction phase, initial subject–predicate–object triples are derived from the preprocessed text, guided by schema-aware prompts and informed by the document's hierarchical context to enhance accuracy. Our pipeline replaces the traditional "Define" phase of EDC with a schema-driven resolution process, thereby reducing our reliance on LLMs within the pipeline. While EDC+R included a dedicated "Define" phase for generating relation definitions via LLM calls, we replaced this with the Resolve step, leveraging schema context. Following the resolution phase, our pipeline performs similarity-based triple merging and redundancy reduction, clustering near-duplicate triples using semantic similarity measures to produce a more precise and coherent knowledge graph. The final canonicalization step then aligns these triples to the domain ontology, ensuring consistent representation of entities and relations across the graph.

5. Implementation challenges and Future work

Currently, we are testing the proposed approach within our pipeline. However, further team brainstorming has raised several complex questions that require deeper investigation. While our current relation-level retrieval and refinement process provides a solid foundation, several challenges persist: Firstly, cosine similarity between relation strings captures only surface-level semantics and may not resolve deeper ambiguities, mainly in domains where relations differ slightly in meaning or usage. Our current pipeline lacks comprehensive integration of entity-type alignment and attribute-level matching, which are crucial for accurately mapping entities into the ontology's class hierarchy, especially when properties or contextual roles distinguish classes.

Future work should aim to effectively extract and manage the full ontology during schema retrieval, incorporating class descriptions, hierarchical paths, and attribute-level metadata into the retrieval and canonicalization stages to improve knowledge graph alignment. Additionally, handling n-ary relations, nested statements, and context-dependent relation semantics remains an open challenge, particularly in long, hierarchical enterprise documents. Finally, rigorous human evaluation involving industry experts and knowledge representation researchers is essential to validate the quality of the resultant triples. Addressing these issues will be central to transforming our current prototype into a robust, industry-grade knowledge graph construction pipeline.

Acknowledgments

This publication is part of the project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21 which is (partly) financed by the Dutch Research Council (NWO).

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT-4, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. Claude Sonnet 4 (GitHub Copilot) was utilized during code development phases to assist the authors. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in neural information processing systems 33 (2020) 9459–9474.

[2] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, arXiv preprint arXiv:2312.10997 2 (2023).

[3] S. Gupta, R. Ranjan, S. N. Singh, A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions, arXiv preprint arXiv:2410.12837 (2024).

[4] Z. Xu, M. J. Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang, Z. Li, Retrieval-augmented generation with knowledge graphs for customer service question answering, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 2905–2909.

[5] S. Kashmira, J. L. Dantanarayana, J. Brodsky, A. Mahendra, Y. Kang, K. Flautner, L. Tang, J. Mars, A graph-based approach for conversational ai-driven personal memory capture and retrieval in a real-world application, arXiv preprint arXiv:2412.05447 (2024).

[6] B. Zhang, H. Soh, Extract, define, canonicalize: An llm-based framework for knowledge graph construction, arXiv preprint arXiv:2404.03868 (2024).

[7] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, K. Lata, Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text, in: International semantic web conference, Springer, 2023, pp. 247–265.

[8] J. Nie, X. Hou, W. Song, X. Wang, X. Zhang, X. Jin, S. Zhang, J. Shi, Knowledge graph efficient construction: Embedding chain-of-thought into llms, Proceedings of the VLDB Endowment. ISSN 2150 (2024) 8097.

[9] A. G. Regino, J. C. Dos Reis, Can llms be knowledge graph curators for validating triple insertions?, in: Proceedings of the workshop on generative AI and knowledge graphs (GenAIK), 2025, pp. 87–99.

[10] X. Feng, X. Wu, H. Meng, Ontology-grounded automatic knowledge graph construction by llm under wikidata schema, arXiv preprint arXiv:2412.20942 (2024).

[11] J. Z. Pan, S. Razniewski, J.-C. Kalo, S. Singhania, J. Chen, S. Dietze, H. Jabeen, J. Omeliyanenko, W. Zhang, M. Lissandrini, et al., Large language models and knowledge graphs: Opportunities and challenges, arXiv preprint arXiv:2308.06374 (2023).

[12] TeX - LaTeX Stack Exchange, Drawing flow diagram in LaTeX using TikZ, https://tex.stackexchange.com/questions/149602/drawing-flow-diagram-in-latex-using-tikz, 2014. Accessed: 2025-09-10.

[13] T. Gan, Q. Sun, Rag-mcp: Mitigating prompt bloat in llm tool selection via retrieval-augmented generation, arXiv preprint arXiv:2505.03275 (2025).