Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 134 (2025) 1125–1130

www.elsevier.com/locate/procedia

Proceedings of the 58th CIRP Conference on Manufacturing Systems 2025

Weaving Knowledge Graphs and Large Language Models (LLMs): Leveraging Semantics for Contextualized Design Knowledge Retrieval

Katharina Theuner[a], Tomas Mikael Elmgren[b], Axel Götling[b], Marvin Carl May[c,1], Haluk Akay[d]

[a]wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76131 Karlsruhe, Germany
[b]KTH Climate Action Centre, KTH Royal Institute of Technology, Stockholm, Sweden
[c]Department of Mechanical Engineering, Massachusetts Institute of Technology (MIT), 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[d]Department of Mechanical Engineering, Delft University of Technology, 2628 CD Delft, The Netherlands

* Corresponding author: Marvin Carl May, Tel.: +49-1523-950-2624. *E-mail address:* mc_may@mit.edu

Abstract

Demographic change in Europe challenges companies as retiring employees take valuable expertise with them. To address this, knowledge graphs (KGs) are emerging as tools for structured knowledge representation. Simultaneously, large language models (LLMs) are increasingly being used as innovative solutions for information retrieval. However, LLMs generally process only public knowledge, and recent approaches integrating Retrieval Augmented Generation (RAG) for private knowledge retrieval often lack contextual relevance. To enhance trustworthiness and overcome these limitations, a method is proposed for embedding latent problem-solving structures within design processes into LLM-driven information retrieval systems. Using a case study in energy infrastructure, a KG of design problems was constructed by extracting functional requirements from semi-structured documentation via LLMs. This KG is further utilized by an LLM to answer queries, with results visualized through an interactive interface. Validation through field studies with engineers underscores the approach's effectiveness in enhancing contextual and trustworthy knowledge dissemination.

Keywords: Knowledge Engineering; Large Language Models; Design; Knowledge Graph

1. Introduction

Design is the human-driven process of identifying problems and developing solutions. Practices of documenting this process can be inconsistent, resulting in unstructured multi-modal artifacts of a creative human experience. When a new designer attempts to build on prior work, they are often limited to naive keyword-matching methods to retrieve knowledge artifacts or existing CAD models [1] representing tremendous time and capital investment. Recently, Large Language Models (LLMs) have been integrated into information retrieval practices at design organizations to break reliance on keyword-matching and semantically connect queries to answers by meaning. These models can process and understand complex information, aiding decision-making and generating results [2]. However, the direct application of such technology prone to positive affirmation can result in a rebound effect for design knowledge preservation. Answers are pushed without context or validation resulting in the user being removed from an educational process.

Coinciding with the increasing usage of generative AI is another trend: the last generation of designers, with professional expertise applying scientific fundamentals to design predating a reliance on computational aids, are retiring. By 2050, 30% of the European population will be older than 65 [3]. This trend poses a significant threat to organizations as employees with more than 25 years of work experience will become a rare resource [4]. Knowledge must be transferred from senior experts to the novice workforce, though this process requires substantial resources [5].

Knowledge must be urgently preserved before experts retire. To address this need, we utilize knowledge graphs (KG) as a framework for structuring information [6]. KGs allow for storing vast amounts of data [7], and structured information can be represented logically [8]. The fundamental structure of a knowledge graph is relational [9, 6] making it ideal for providing functionally relevant design context. While the develop-

2212-8271 © 2025 The Authors. Published by Elsevier B.V.

10.1016/j.procir.2025.03.073

ment of KGs based on structured data is established, a research gap remains concerning integrating unstructured data [10].

LLMs are a powerful tool for semantically extracting information from unstructured texts. However, data on which LLMs are trained are general language examples and do not include knowledge for specific use cases [11], let alone design and manufacturing [12]. To address this challenge, there has been growing research interest in combining context specific KGs with LLMs [10, 8]. A key challenge is that KGs remain reliant on structured datasets, which can lead to the loss of valuable context when KGs are built from unstructured documents.

To overcome the challenges this work proposes a framework utilizing an LLM to extract data from semi-structured documents to construct a knowledge graph of organizational-level problem-solving knowledge for a case study in energy infrastructure design. The LLM extractions are validated through a field study with both novice and expert design professionals. Finally a user interface was designed to present retrieved information to users together with the path in the knowledge graph showing context and functionally-adjacent sources.

This work is structured as follows. Section 2 describes the recent research regarding RAG, Section 3 provides a description of the proposed framework. Section 4 show a validation by applying a field study. The discussion is provided in Section 5, the work closes with a summary and outlook in Section 6.

2. State-of-the-art

When LLMs cannot access needed data, they invent false information [11]. This is particularly the case when private or domain-specific knowledge is considered. To overcome this issue, models could undergo retraining or get fine-tuned, but this is a work-intensive process resulting in high costs [8] and even impossible for models that are only accessible through an Application Programming Interface (API) [13]. Another approach that has recently been discussed is the implementation of RAG, which can be applied when LLMs lack domain-specific knowledge [11, 8]. Data from external knowledge sources can be retrieved, such as databases or tools. LLMs access this stored information through prompts or embeddings [14].

Due to the attention that is currently posed on AI, knowledge graph-enhanced large language models is a recently discussed topic in literature, although the number of published work is still limited [13, 14].

Table 1 shows an overview of the literature illustrated below. In the approach from [15], an LLM accesses a knowledge graph. The user question and the information from the KG are compared using cosine similarity. The given answer is based on the information retrieved that achieves the highest values. Although the answers that could be offered are detailed and provide context, the approach is work-intense due to the necessary training. In other methods, more attention is put on the semantics of the KG [16], or instead of providing more information via a knowledge graph, a RAG like procedure is used to verify data provided by the LLM [17]. Both approaches [17, 16] provide more context and can answer more detailed questions, but the level of detail is still perceived as medium.

Table 1. Overview of research papers addressing RAG

Approach	Degree of context	Training	Purpose
[15]	●	X	QA
[16]	◐	X	QA
[17]	◐	-	check information
[18]	◐	X	QA
[10]	○	X	QA
[19]	○	X	QA
[11]	○	-	QA
[20]	○	X	QA
[21]	-	X	suggestions, identification

Legend: ●high degree, ◐medium degree, ○low degree, QA = question-answer, X = apply, - = does not apply. Source: Own representation.

[10] propose an approach within a user question is proceed to an OpenAI GPT4-endpoint. It is converted into a query statement to retrieve knowledge from a graph on which the response is built. Although the graph is constructed in the manufacturing context based on unstructured and structured data, existing ontologies are used. Four nodes and relation labels define the graph. Within the work of [19], a question is already linked to five potential answer choices. In a context node, this question-answer representation is stored within the KG with the help of a language model. A joint working graph is built by linking it to the entities describing various topics. The relation between the context node and all other nodes described within the KG is determined with a relevance score. Within the approach of [11], an LLM is linked with a Neo4j knowledge graph. Although the information base is in Japanese, a robot can chat in English about the cooking knowledge stored within the knowledge graph. [20] aim to consider both structured and unstructured knowledge, unify, and then use it to answer a user question utilizing a retriever-reader architecture where different information sources are first converted into text, and a retriever selected appropriate passages from these texts. A reader processes the various parts together to answer the user's question. Although all aforementioned approaches [11, 10, 20, 19] can answer questions, they do not provide much context. Only simple questions with answers of a few words are described.

Within the approach of [21], a GPT-3.5 model is fine-tuned for intelligent aircraft maintenance by integrating an aircraft structure ontology. Therefore, the model increases its performance by identifying aircraft components and suggesting maintenance actions. In contrast to the aforementioned approaches, it does not focus on question-answering tasks.

Although a few approaches exist that provide more context, in most cases, training is needed that is work intensive and costly [22]. Moreover, although general semantic comparisons are used for providing context-specific knowledge, a more in-depth comparison is desirable for providing relevant and connected design information on a higher level. Therefore, resource-friendly approaches are desired. Using KGs could solve this problem, and, hence, this research investigates if KGs can build the knowledge base for LLMs to sufficiently support design knowledge retrieval. This work follows an approach of using an LLM to construct a knowledge graph [8] and to query the design knowledge without LLM re-training.

3. Framework

The presented methodology is divided into three sections. First, Subsection 3.1 creates a KG with the assistance of a LLM. Then, Subsection 3.2 leverages the information stored in the knowledge graph to augment the question-answering process. Finally, a user interface is presented in Subsection 3.3.

3.1. Knowledge Graph Construction

The key objective for developing a design knowledge graph is to represent the problem-solving process utilizing relational connections between nodal entities. Assuming the information source to be a corpus of design documents, the resulting design knowledge graph contains the source document as a central entity. In order to preserve the functional structure of the documented problem-solving, problems addressed by the document constitute the radial nodes. These problems buried in documentation were semantically extracted from the corpus using an LLM. In implementation, the OpenAI model gpt4o-mini [23] is used as it is considered a state-of-the-art transformer-based model. A prompt was designed to guide the model in extracting problems. The behavior and roles of both the system and the user are described in the message. A specific scenario is outlined, in which the model is asked to extract problems from provided texts that could be considered relevant issues for a novice engineer without prior knowledge, ensuring that the answers can be found within the text. This is outlined in the system role. The prompt includes a variable referencing the relevant text documents stored in the knowledge graph (KG). The user's role focuses on searching for questions within the relevant texts.

Reliance purely on problem nodes can omit valuable background information pertinent to user questions requiring detailed context e.g. step-by-step work processes rather than merely tool names. Hence, an additional text node is integrated into the KG to represent the full text in a single entity. Detailed answers contained within this full text node are connected to questions which are most similar to the design problem nodes acting as labels for the document cluster as visualized in Figure 1. The graph database library is utilized to implement the final knowledge graph connecting the extracted functional information representing the design process.

3.2. Knowledge Graph-based LLM Application

The key metric by which answers are connected to queries is semantic similarity, estimated by the cosine similarity between LLM-generated vector embeddings representing language meaning. Using the text-embedding-3-large model from OpenAI [23], embeddings of the user question and all problem nodes represented in the KG are generated. To compare both on a semantic level, the cosine similarity value is calculated. According to predefined rules selected in a parametrization phase, a certain number of problem nodes is chosen and the corresponding text nodes based on the calculated cosine similarity values. The model is then instructed to generate an answer to a question based on the text nodes linked to the chosen problem nodes. To enhance this process, we add an additional knowledge retrieval loop. This involved creating summary nodes, condensed representations of the text nodes, to explore new pathways not identified in the initial knowledge retrieval round. A semantic comparison is then conducted between the summary node linked to the problem node with the highest cosine similarity and all other summary nodes. The node with the highest similarity is selected to provide supplementary information.

The system instructions are formulated to minimize incidents of false information generated when the LLM cannot retrieve information with confidence, known as a "hallucination". This will be indicated if the LLM fails to provide a response based on the supplied documents. No extensive training is needed to reach the desired model behavior. This could solely be achieved by varying the prompts and explain detailed the desired model behavior.

3.3. User Interface

A key aim of the design knowledge retrieval system presented in this paper is not only to efficiently map queries to answers but also to show the user how the answer was found. Without transparency, user trust is impacted. Without context, user comprehension is impacted. The system is intended to represent organizational-level quantities of knowledge, resulting in large graphs from which answers are retrieved. This creates a conflict between prioritizing transparency and overwhelming the user. Another aim of the system is to facilitate a design research journey such that the ease of finding answers does not rob the user of inadvertent discovery of related knowledge. Simultaneously, the system must prioritize presentation of accurate, focused results from the knowledge graph source.

The UI is designed for interaction with the system on a desktop pc, illustrated in Figure 2. The left half of the space is a stack of three text-based modules, and the right half is reserved for an interactive graphic visualization. A text field below the prompt "Ask a Question" invites the user to enter a query, which is passed as an embedding according to the method in Subsection 3.2 to obtain a list of answers, the first of which is returned in the module immediately below. Accompanying the answer is an information module listing the linked source document used to abstract the answer, the node type, a preview of the full context, and finally a similarity score corresponding to the semantic similarity computed between the query and answer.

The visualization on the right updates to display a grayed-out representation of the complete knowledge graph. Superimposed is a "Knowledge Path" which is a highlighted sequence of node connections reflecting organizational structure of the source documents, similar to a file path. The user can discover functionally similar problem descriptions by navigating to nodes nearby in the knowledge graph's semantic space.

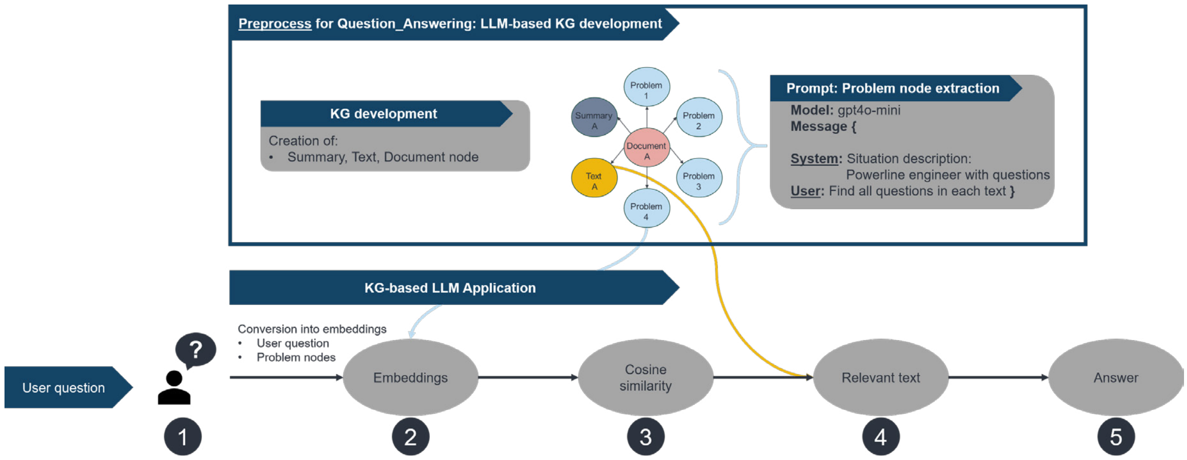

Fig. 1. Detailed LLM Application for Design Knowledge Graph creation

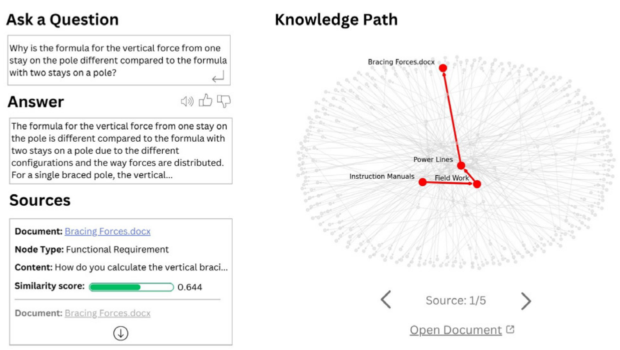

Fig. 2. User interface for returning answers and transparently showing path to relevant prior problem nodes represented in the knowledge graph representation of the documentation source.

For every query, a list of *N* most relevant answers which achieve a predefined semantic similarity threshold are returned. The user can navigate down this list using arrows on both the source text-based module and the graphic visualization. This user interface was implemented using the Streamlit framework to connect the graph database to a web application.

4. Results

Documents from a design consultancy in the energy infrastructure sector were used as a case study to implement the framework. The documents focused on the design of structural systems for electric power transmission. They encompassed various supportive information for novice engineers, such as general work instructions. Some had a clear structure and described a step-by-step process, others were a summary of guidelines and general suggestions. The provided documents were used to build the knowledge graph. Nine different documents were included, and 148 problem nodes could be extracted.

The key aim of this validation analysis was to evaluate the reliability of the system for returning accurate and relevant answers to users. Given the methodology relies on the veracity of extractions provided by an LLM, a field study was conducted to benchmark the correctness of the answers provided by the LLM in comparison to human professionals' assessment of ground truth. From the design consultancy upon which the case study was performed, we recruited 6 junior designers and 2 expert designers. The junior designers were given access to an instructional document, and asked to answer 5 questions. The document contained textual descriptions, formulas, and calculation best practices for the structural design of a power transmission pole. In addition, as a baseline another approach was used in which the user question was semantically compared to documents, based on which the answer was given.

Next, the two expert designers were asked to rank all answers according to correctness. For every question, there was disagreement between the expert designers as to the rankings, which differed for one question for a range of rank 1 to rank 6. This demonstrated the challenge of identifying a ground truth in a creative domain such as design from which to establish a benchmark. However their answers and rankings were used to compare the methodology described in this paper to the answers achieved by junior experts and an LLM.

Therefore all questions were integrated in the proposed framework and based on the previously mentioned answers and rankings it could be validated. Across all five questions, no incorrect information was provided. However, a distinction must be made between correctness and the level of detail. Wherease the first ensures that no false information is provided, the level of detail focuses on including all relevant aspects without omission. Three answers demonstrated an adequate degree of detail. One response was impacted by the complexity of interpreting formulas, which posed a challenge. For the final question, the level of detail could be improved by slightly rephrasing the question to better guide the response. In addition by using the summary nodes further information could be provided for one question leading to information which could not be retrieved in the first round of knowledge acquisition. In total, the answers provided by our method were on par or exceed that of junior engineers. One main aspect of this framework is the detailed search for the correct text fragments important to answer the

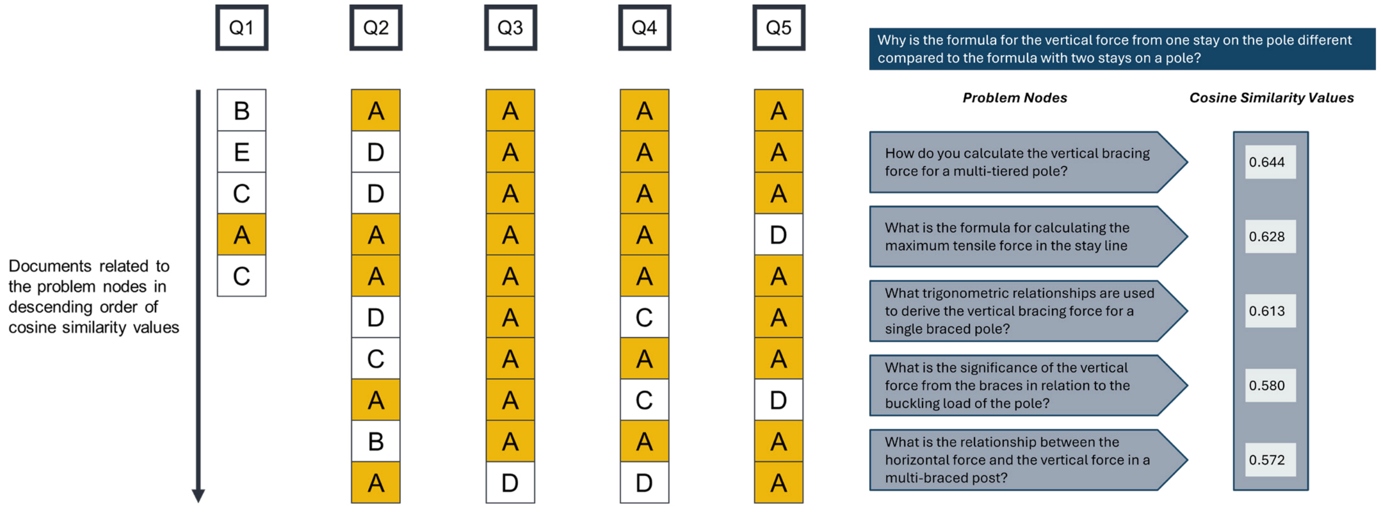

Fig. 3. Cosine similarity values between problem nodes and user questions for field study (left) and an example of different problem nodes for a user question with their corresponding cosine similarity values (right).

questions by using the problem nodes. Semantic similarity can be estimated by the cosine similarity between feature vectors representing text obtained from an LLM. A value closer to 1 relates to high similarity while a value closer to zero indicates low similarity [24]. Figure 3 shows the chosen documents for the five questions asked in the field study. In this case, document A contained the useful information for answering the question. All problem nodes that are marked are linked with the correct text node A for answer generation, all other problem nodes that are not highlighted are linked to other documents not relevant for question answering. The number of problem nodes displayed for each question varies based on the cosine similarity threshold applied to determine their relevance and inclusion in the process. Although the cosine similarity values can differ they are a useful metric to link the correct document to a user question. However, the exact value does not provide much information regarding the correctness of a certain answer. A value that appears low still leads to the correct problem node and corresponding text for answering the question, as it remains relatively higher compared to irrelevant nodes.

5. Discussion

This work proposes a framework for knowledge preservation combining large language models and knowledge graphs. It is developed and validated based on the industrial use case of a consultancy established in the sector of power line engineering. With the rise of digital documentation and LLM technology, the way knowledge is retrieved is evolving. While keyword-based retrieval offers clear query-to-answer connections, LLM-based methods often lack this clarity, functioning as ”black boxes.” This work aims to enhance transparency by representing design knowledge as a graph, linking queries to answers through semantic positions. In addition, the search for the relevant texts by the problem nodes and further information through the summary nodes it ensures an in depth retrieval. While excelling in application a key benefit is that ability to create a long term knowledge base with the KG that is accessed through the LLM. The proposed method additionally does not require expensive and slow retraining of the LLM.

Design documentation often includes non-textual elements like sketches and 3D models. Our method primarily handles textual data, posing limitations for multi-modal integration. As seen during the validation, formulas can be handled to a satisfying degree, however, room for improvement and the ability to cope with different variables still exists. Future work could incorporate nodes for non-textual data, addressing this challenge.

A key advantage of the framework is its adaptability to various contexts, as it relies primarily on prompt engineering and the nexus of the constructed KG and used LLM. This flexibility allows the knowledge graph to be constructed for applications such as manufacturing tasks and maintenance jobs. A challenge facing this research involves evaluating and selecting the appropriate language model for application, given frequent turnover and the often closed-source nature of this software. A further area of research with significant benefits of the proposed method is on the interface of design and manufacturing, i.e. Product-Production-CoDesign [25]. Additionally, frequent changes and the ability to grow the knowledge persistently, e.g. by using the ever increasing digital thread [26], exists.

6. Summary and Outlook

The demographic shift in Europe presents challenges, particularly in knowledge preservation, as companies face numerous retirements. Within the setting of design, validated with design of energy power transmission systems, our approach addresses that gap. This work, hence, proposes a framework combining knowledge graphs and large language models focusing on dual enhancement, where the KG is built using LLMs, which, in turn, leverage the KG for context-specific answers. This approach is efficient in gaining deeper insights and discovering connections through advanced KG searches. The system also benefits senior staff by facilitating access to comprehensive information. The simplicity of prompt engineering makes the method replicable across companies and sectors, offering broader applicability. Future research should address the inclusion of multi-modal data and the enhanced processing of formulas. Additionally, the dynamics of real manufacturing systems, beyond design, for instance during adaptions and contingencies, should be regarded and the approach expanded.

Acknowledgements

The Federal Ministry of Education and Research (BMBF) is funding the "Resipro" project "FENI-X: Flexibility and efficiency in production networks – Decisive response to X" managed by the Project Management Agency Karlsruhe. The authors are responsible for the content of this publication. Additionally funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—MA 11729/1.

References

[1] C. Krahe, M. Marinov, T. Schmutz, Y. Hermann, M. Bonny, M. May, G. Lanza, Ai based geometric similarity search supporting component reuse in engineering design, Procedia CIRP 109 (2022) 275–280. doi:10.1016/j.procir.2022.05.249.

[2] Y. Xia, M. Shenoy, N. Jazdi, M. Weyrich, Towards autonomous system: flexible modular production system enhanced with large language model agents, in: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), 2023, pp. 1–8, arXiv:2304.14721 [cs, eess]. doi:10.1109/ETFA54631.2023.10275362.

[3] European Union, The impact of demographic change in a changing enironment, accessed: 2024-09-12 (2023).
URL https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/new-push-european-democracy/impact-demographic-change-europe_en

[4] E. Fenoglio, E. Kazim, H. Latapie, A. Koshiyama, Tacit knowledge elicitation process for industry 4.0, Discover Artificial Intelligence 2 (1) (2022) 6. doi:10.1007/s44163-022-00020-w.

[5] S. K. Freire, S. S. Panicker, S. Ruiz-Arenas, Z. Rusák, E. Niforatos, A Cognitive Assistant for Operators: AI-Powered Knowledge Sharing on Complex Systems, IEEE Pervasive Computing 22 (1) (2023) 50–58. doi:10.1109/MPRV.2022.3218600.

[6] J. Yan, C. Wang, W. Cheng, M. Gao, A. Zhou, A retrospective of knowledge graphs, Frontiers of Computer Science 12 (1) (2018) 55–74. doi:10.1007/s11704-016-5228-9.

[7] X.-S. Yang, S. Sherratt, N. Dey, A. Joshi (Eds.), Proceedings of Fifth International Congress on Information and Communication Technology: ICICT 2020, London, Volume 2, Vol. 1184 of Advances in Intelligent Systems and Computing, Springer Singapore, Singapore, 2021. doi:10.1007/978-981-15-5859-7.

[8] J. Z. Pan, S. Razniewski, J.-C. Kalo, S. Singhania, J. Chen, S. Dietze, H. Jabeen, J. Omeliyanenko, W. Zhang, M. Lissandrini, R. Biswas, G. de Melo, A. Bonifati, E. Vakaj, M. Dragoni, D. Graux, Large Language Models and Knowledge Graphs: Opportunities and Challenges, DROPS-IDN/v2/document/10.4230/TGDK.1.1.2Publisher: Schloss-Dagstuhl - Leibniz Zentrum für Informatik (2023). doi:10.4230/TGDK.1.1.2.

[9] M. C. May, L. Kiefer, A. Kuhnle, G. Lanza, Ontology-based production simulation with ontologysim, Applied Sciences 12 (3) (2022). doi:10.3390/app12031608.

[10] Y. Li, B. Starly, Building a knowledge graph to enrich ChatGPT responses in manufacturing service discovery, Journal of Industrial Information Integration 40 (2024) 100612. doi:10.1016/j.jii.2024.100612.

[11] K. Jokinen, G. Wilcock, Exploring a Japanese Cooking Database, in: Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, ACM, Boulder CO USA, 2024, pp. 578–582. doi:10.1145/3610978.3640622.

[12] M. C. May, J. Neidhöfer, T. Körner, L. Schäfer, G. Lanza, Applying natural language processing in manufacturing, Procedia CIRP 115 (2022) 184 – 189. doi:10.1016/j.procir.2022.10.071.

[13] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, Y. Shoham, In-context retrieval-augmented language models, Transactions of the Association for Computational Linguistics 11 (2023) 1316–1331.

[14] L. Yang, H. Chen, Z. Li, X. Ding, X. Wu, Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling, IEEE Trans. on Knowledge and Data Engineering (2024).

[15] L. Xu, L. Lu, M. Liu, C. Song, L. Wu, Nanjing yunjin intelligent question-answering system based on knowledge graphs and retrieval augmented generation technology, Heritage Science 12 (1) (2024) 118. doi:10.1186/s40494-024-01231-3.

[16] M. R. A. H. Rony, R. Usbeck, J. Lehmann, DialoKG: Knowledge-structure aware task-oriented dialogue generation, in: Findings of the Association for Computational Linguistics: NAACL 2022, Association for Computational Linguistics, pp. 2557–2571. doi:10.18653/v1/2022.findings-naacl.195.

[17] M. Mountantonakis, Y. Tzitzikas, Using multiple RDF knowledge graphs for enriching ChatGPT responses, in: G. De Francisci Morales, C. Perlich, N. Ruchansky, N. Kourtellis, E. Baralis, F. Bonchi (Eds.), Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track, Springer Nature Switzerland, pp. 324–329. doi:10.1007/978-3-031-43430-3_24.

[18] T. Ashby, B. K. Webb, G. Knapp, J. Searle, N. Fulda, Personalized quest and dialogue generation in role-playing games: A knowledge graph- and language model-based approach, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, ACM, 2023, pp. 1–20. doi:10.1145/3544548.3581441.

[19] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec, QA-GNN: Reasoning with language models and knowledge graphs for question answering, in: Proceedings of the 2021 Conf. of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 535–546. doi:10.18653/v1/2021.naacl-main.45.

[20] B. Oguz, X. Chen, V. Karpukhin, S. Peshterliev, D. Okhonko, M. Schlichtkrull, S. Gupta, Y. Mehdad, S. Yih, UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering, in: Findings of the Association for Computational Linguistics: NAACL 2022, Association for Computational Linguistics, 2022, pp. 1535–1546. doi:10.18653/v1/2022.findings-naacl.115.

[21] P. Wang, J. Karigiannis, R. X. Gao, Ontology-integrated tuning of large language model for intelligent maintenance 73 (1) (2024) 361–364. doi:10.1016/j.cirp.2024.04.012.

[22] V. Liu, Y. Yin, Green AI: exploring carbon footprints, mitigation strategies, and trade offs in large language model training 4 (1) 49. doi:10.1007/s44163-024-00149-w.

[23] OpenAI, accessed: 2024-08-21 (2024). [link].
URL https://platform.openai.com/docs/models

[24] N. B. Korade, M. B. Salunke, A. A. Bhosle, P. B. Kumbharkar, G. G. Asalkar, R. G. Khedkar, Strengthening sentence similarity identification through openai embeddings and deep learning., International Journal of Advanced Computer Science & Applications 15 (4) (2024). doi:10.14569/IJACSA.2024.0150485.

[25] M. C. May, L. Schäfer, A. Frey, C. Krahe, G. Lanza, Towards product-production-codesign for the production of the future, Procedia CIRP 119 (2023) 944–949. doi:10.1016/j.procir.2023.02.172.

[26] H. Akay, S. H. Lee, S.-G. Kim, Push-pull digital thread for digital transformation of manufacturing systems, CIRP Annals 72 (1) (2023) 401–404. doi:10.1016/j.cirp.2023.03.023.