Intelligent Human Systems Integration (IHSI 2025), Vol. 160, 2025, 765–775
https://doi.org/10.54941/ahfe1005883

Automatic Creation of Assembly Instructions by Using Retrieval Augmented Generation

Robin Herbort, Dominik Green, and Sven Hinrichsen

Ostwestfalen-Lippe University of Applied Sciences and Arts, Lemgo, 32657, Germany

ABSTRACT

The application of Large Language Models (LLMs) for the automated generation of assembly instructions shows significant potential for improving work preparation in production processes. However, challenges remain regarding the overall information quality and precision of the generated instructions. In light of these challenges, this study explores how the information quality of automatically generated assembly instructions can be enhanced through the targeted provision of structured input data, such as Assembly and Quantity BOMs (Bills of Materials), as well as the use of optimized prompt chaining techniques. The methodology employs ChatGPT-4o in combination with Retrieval Augmented Generation (RAG) within the Microsoft Azure environment. The results demonstrate that structured data inputs, particularly the use of Assembly BOMs with defined Tool-to-Component relations, significantly improve the precision and relevance of the generated instructions. Despite these advancements, achieving consistent information quality remains a barrier to broader practical implementation. Therefore, feedback loops should be integrated into the assembly instruction generation process to ensure continuous refinement and reliability. Future research should investigate the use of RAG or similar frameworks, focusing on optimizing data structures and implementing feedback mechanisms to enhance the automated generation of assembly instructions.

Keywords: Retrieval augmented generation, Large language model, Assembly instructions

INTRODUCTION

The trend towards product configuration by customers, as well as the integration of increasingly more functions into products, combined with shorter production cycles, has led to a significant increase in complexity (Schuh et al., 2017) and, consequently, to a substantial rise in the amount of information that companies have to process. Every customer order in variant-rich series production, and especially in industrial single-piece production, entails specific requirements. These requirements impact the entire product development and order processing chain. For the role of the industrial engineering department, this trend means an increase in administrative efforts to provide assembly workers with a variety of assembly instructions tailored to their needs. Therefore, new approaches are required to manage this complexity and keep complexity-related costs low (Hvam et al., 2020).

One approach involves the partial automatic generation of assembly instructions using Retrieval Augmented Generation (RAG). A software system like RAG combines the retrieval of information from a database (Information Retrieval) with the use of a Large Language Model (LLM). To determine the state of research regarding the use of the RAG approach in the production context, a systematic literature review (SLR) was conducted. For this, the multidisciplinary database "Web of Science" was used, applying the search terms "LLM" AND "Retrieval Augmented Generation" AND "Manufacturing" in the fields "searches title," "abstract," "keyword plus," and "author keywords." This approach led to the identification of three relevant articles. An additional search was conducted through Google Scholar using the search terms "Retrieval Augmented Generation" and "Manufacturing." This search resulted in the identification of five further relevant sources (see Table 1). However, no source was identified in which RAG is used for the generation of assembly instructions.

Table 1. Publications identified via a literature review.

No.	Author	Description
[1]	Chandrasekhar et al., 2024	Model for querying material data in additive manufacturing
[2]	Buehler, 2024	Application for analysis in material mechanics
[3]	Xia et al., 2024	Creation of digital twins in a standardized format
[4]	Machado 2024	Development of an assistant for industrial maintenance
[5]	Bahr et al., 2024	Standardization or arguments in a Failure Modes and Effects Analysis
[6]	Álvaro, 2024	Method for cause/solution search for potential manufacturing defects
[7]	Freire et al., 2024	Model for knowledge sharing in production
[8]	Freire et al., 2023	Creation of cognitive assistants in production

Nevertheless, one study exists in which an assembly instruction was automatically generated using LLMs. In this study, Meyer et al. (2024) conducted experiments in which an instruction for mounting a pneumatic assembly was automatically created. This assembly instruction was to include a listing of assembly tasks while considering the assembly sequence. Additionally, instructions for the use of tools and adherence to quality guidelines were to be provided. In the experimental series, three different LLMs – ChatGPT-3.5 Turbo, ChatGPT-4, and ChatGPT-4V – were used. Three different prompt techniques – Zero-Shot, Context, and Prompt Chaining – were applied. The input data consisted of a Quantity Bill of Materials (BOM), an Assembly BOM, and an Assembly BOM with additional information regarding the sequence of individual assembly steps. Thus, three

independent variables with three levels each were examined, resulting in a 3x3x3 factorial experimental design with a total of 27 experiments (see Figure 1).

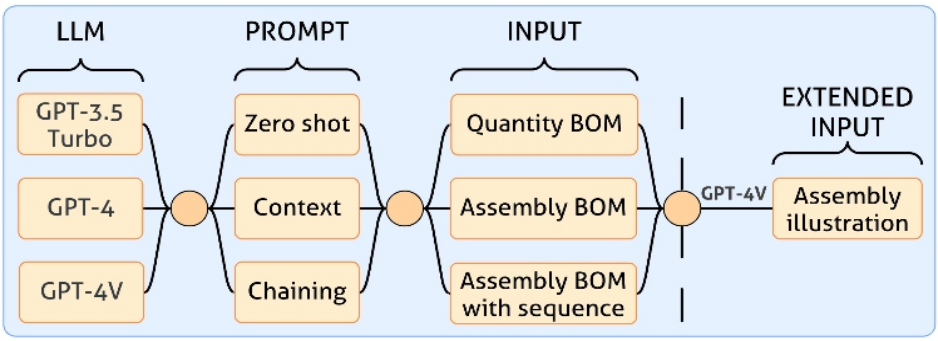

Figure 1: Experimental design of the study by Meyer et al. (2024).

In the case of ChatGPT-4V, an image of the pneumatic assembly was additionally provided to the LLM, for which the assembly instruction was to be generated (see Figure 2). This assembly was selected because it consists of standard components, for which extensive information is available online. It was therefore hypothesized that the language model would incorporate this information into its processing.

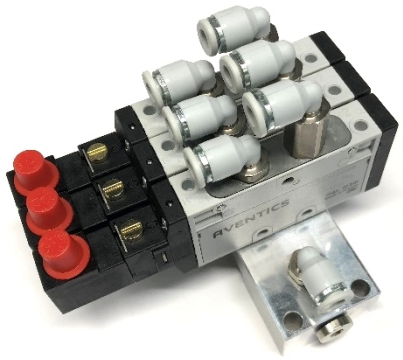

Figure 2: Representation of the provided assembly for the experimental procedure.

In the study, the dependent variable was the information quality of the generated assembly instruction (output). This was operationalized using six criteria (see Table 2). A four-point Likert scale was applied to evaluate each criterion: (1) criterion not met, (2) criterion mostly not met, (3) criterion mostly met, (4) criterion fully met. The assessment of the output based on these criteria was conducted separately for information regarding the assembly process, the tools to be used, and the quality assurance measures.

The comparison of the three LLM versions – ChatGPT-3.5 Turbo, ChatGPT-4, and ChatGPT-4V from OpenAI – revealed differences in terms

of information quality. The most recent LLMs from OpenAI, ChatGPT-4 and ChatGPT-4V, achieved higher information quality than the older version, ChatGPT-3.5 Turbo. These differences are likely due to improved processing of contexts and relevant data in the newer versions. Furthermore, it was observed that the application of prompt chaining resulted in the highest information quality, whereas zero-shot and context prompting produced lower levels of information quality.

Table 2. Criteria for evaluating the information quality of the assembly instructions.

No.	Criteria	Description
(1)	Appropriate amount of data	The amount of information is sufficient for an inexperienced employee to assemble the module completely and correctly.
(2)	Completeness	All relevant information and steps are fully included, allowing assembly to be completed without additional sources or inquiries.
(3)	Concise representation	The information is presented clearly and concisely.
(4)	Free of error	The instruction does not contain any errors.
(5)	Understandability	The instructions are easy to understand.
(6)	Appropriate sequence	The assembly steps are generated in the correct sequence.

Regarding the input data, the highest information quality was achieved by providing the Quantity BOM in combination with the LLM ChatGPT-4V and the prompt-chaining technique. The study concluded that while LLMs have potential for the automatic generation of assembly instructions, the overall low information quality of the output represents a significant barrier to practical application.

Objective of This Study

The study presented in this paper aims to improve the information quality of the generated assembly instructions by employing a modified experimental design. To ensure comparability with the initial study by Meyer et al. (2024), the experimental design described in the following section is aligned with that study. The new experimental design is based on the following key hypotheses:

Prompt Chaining: A prompt-chaining method that is reduced to essential statements achieves higher information quality in the assembly instructions than a method that provides a large amount of information for each work step.

Bills of Materials (BOMs): The use of RAG and ChatGPT-4o enables more efficient processing of structured data. Compared to the Quantity BOM, the use of the Assembly BOM leads to higher information quality, as the structured provision of data in the form of an Assembly BOM supports more precise and context-aware generation of instructions.

Assembly Sequences: Providing assembly sequences (specifying the order of assembly steps) leads to higher information quality in the assembly instructions compared to cases where no assembly sequences are provided.

Tools: Providing Tool-to-Component relations results in higher information quality of the assembly instructions compared to not providing this information, meaning generating it from the data set of ChatGPT-4o.

Best Result: The best result of this study surpasses the best result of the study by Meyer et al. (2024) in terms of the information quality of the generated assembly instructions. The "best result" is defined as the combination of independent variables that, on average, leads to the highest information quality.

METHOD

The experimental setup can be described in terms of the phases: input, processing, and output. The input data for the various experimental runs are stored as PDF files in a database. The processing, in contrast to the experiments by Meyer et al. (2024), is carried out using RAG. This approach is implemented through Microsoft Azure. The output represents the system's response in the form of a generated assembly instruction and is evaluated based on the six criteria (see Table 2).

Input

To ensure the comparability of results between this study and the initial study by Meyer et al. (2024), the same assembly was used for the automatic generation of assembly instructions as in the first study (see Figure 2). The present study is based on a 2x2x2x2 factorial experimental design (see Figure 3). First, a distinction is made between the prompt-chaining method used by Meyer et al. (2024) and an optimized prompt-chaining method. The optimized method is focused on essential information and is specifically adapted to the input data. Second, a distinction is made between the Quantity BOMs and the Assembly BOMs. Third, in half of the experiments, information on the assembly sequence is provided, whereas in the other half, it is not. Fourth, in half of the experiments, information on the tools-to-component relations is supplied, while in the other half, this information is omitted. In total, this results in 16 different test settings.

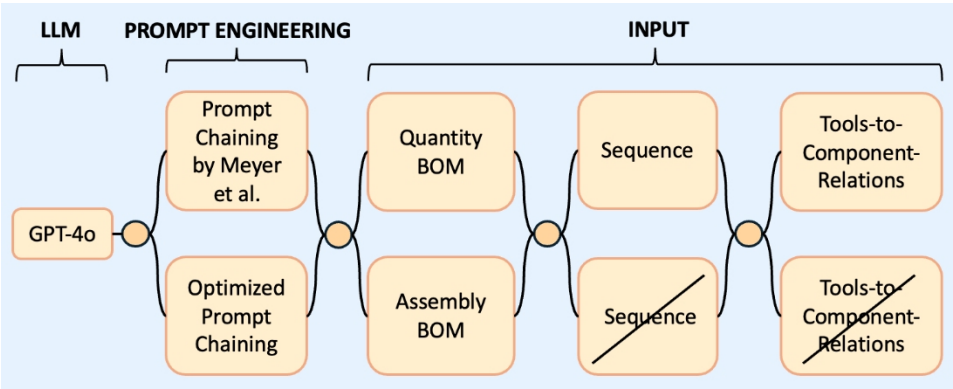

Figure 3: Experimental design of this study.

Processing

The data processing is carried out through the implementation of a RAG Process (Gao et al., 2023), where ChatGPT-4o (LLM), Azure Cognitive Search (Retriever), and an external database (Blob Storage) are integrated within Microsoft Azure (see Figure 4). By linking these RAG resources, domain-specific datasets can be incorporated into the generation of assembly instructions.

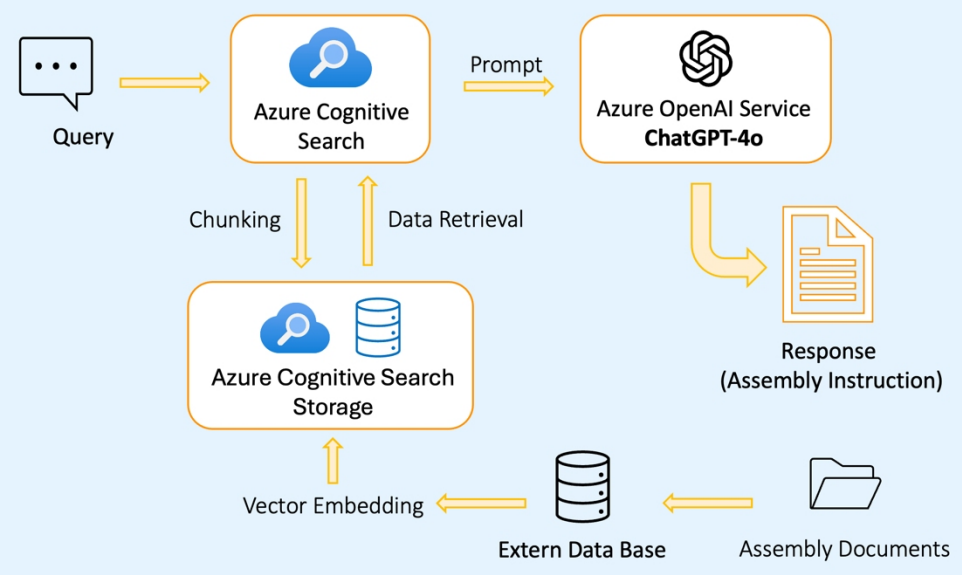

Figure 4: Representation of the RAG process from user input to the system output of the assembly instructions.

The user's query will be vectorized by the retriever (Azure Cognitive Search) and compared with the vectorized and embedded data from the external database. Subsequently, relevant data from the database (Azure Cognitive Search Storage) are retrieved and, along with the user's query, provided to the LLM. The LLM then generates an assembly instruction based on the external data, the user's query, and its own data pool (Lewis et al., 2020).

Output

The output includes the generated step-by-step assembly instruction. It consists of three components: First, the individual work steps are explained (Assembly Notes). Second, the tools to be used are listed (Tool Notes), and third, the actions to ensure quality are specified (Quality Notes).

RESULTS

The results of the experiments are presented in an evaluation matrix, known as a heatmap. The heatmap combines numerical values with color codes, which simplifies the interpretation of the results and allows for comparison

with the test results of Meyer et al. (2024). Accordingly, a score of "4" (criterion fully met) is highlighted in green, while a score of "1" (criterion not met) is marked in red. A total of 16 experiments were conducted. The quality of the assembly instructions was evaluated in terms of "Tool Notes," "Assembly Notes," and "Quality Notes" using the six criteria (see Table 2) and a four-point Likert scale (see Figure 5). To assess the influence of the different independent variables on information quality, average scores were calculated. In eight experiments, the prompt chaining method from Meyer et al. (2024) was used, while the remaining eight employed an optimized version of the prompt chaining method.

Experimental							Key elements of assembly instructions																		
Nr.	LLM	Prompt Chaining	Input				Appr. Amount of data			Completeness			Concise representation			Free of error			Understandability			Appropriate sequence			ø
			Q-BOM	A-BOM	SQ	T-t-C	Tool notes	Assembly notes	Quality notes	Tool notes	Assembly notes	Quality notes	Tool notes	Assembly notes	Quality notes	Tool notes	Assembly notes	Quality notes	Tool notes	Assembly notes	Quality notes	Tool notes	Assembly notes	Quality notes	
1	ChatGPT-4o	First Test Series	x				2	2	3	2	2	3	3	2	3	2	2	3	2	2	3	2	1	3	2,3
2		First Test Series	x		x		2	2	3	2	2	3	3	2	3	2	2	3	2	2	3	2	2	3	2,4
3		First Test Series	x			x	2	2	3	2	2	3	3	2	3	2	2	3	2	2	3	2	1	3	2,3
4		First Test Series	x		x	x	3	2	3	2	2	3	3	2	3	2	2	3	2	2	3	2	1	3	2,4
5		First Test Series		x			3	3	3	2	2	3	3	3	3	1	1	3	3	2	3	2	2	3	2,5
6		First Test Series		x	x		3	3	3	2	3	3	4	4	4	1	3	3	3	2	3	2	3	3	2,9
7		First Test Series		x		x	3	2	3	4	3	3	4	4	3	4	3	3	3	2	3	3	2	3	3,1
8		First Test Series		x	x	x	3	3	3	3	2	3	4	3	4	2	2	3	3	2	3	1	2	3	2,7
9		Optimized	x				2	2	3	2	2	3	3	2	3	2	2	3	2	2	3	2	1	3	2,3
10		Optimized	x		x		2	1	3	2	1	3	2	2	3	2	2	3	2	2	3	2	1	3	2,2
11		Optimized	x			x	3	2	4	2	2	3	3	4	4	2	1	3	2	2	3	1	1	3	2,5
12		Optimized	x		x	x	3	2	4	3	1	3	3	3	4	3	2	3	3	2	3	2	1	3	2,7
13		Optimized		x			2	2	3	2	2	2	4	4	4	2	2	3	2	2	3	2	2	3	2,6
14		Optimized		x	x		2	3	3	2	3	2	4	4	3	2	3	3	2	3	3	2	3	3	2,8
15		Optimized		x		x	3	3	4	3	3	3	3	4	3	3	3	3	3	2	3	4	3	3	3,1
16		Optimized		x	x	x	3	4	3	4	3	3	4	4	3	3	3	3	4	3	3	3	3	3	3,3

Q-BOM: Quantity BOM
A-BOM: Assembly BOM
SQ: Sequences
T-t-C: Tool-to-Component-Relations

Dimensions for information quality

Figure 5: Evaluation matrix for the documentation of the test results.

The method from Meyer et al. achieved an average information quality score of 2.6 points across the eight trials, while the optimized method slightly improved the score to 2.7 points. Analog calculations were made for the other three independent variables. As shown in Table 3, the type of BOMs had the most significant effect on information quality. Using a Quantity BOM yielded an average score of 2.4, while the Assembly BOM improved the score by 0.5 points. The second most influential factor was the provision of Tools-to-Component (T-t-C) relations. When this information was not provided, the information quality dropped by an average of 0.3 points.

Table 3. Performance comparison of the test results.

Prompt Chaining	Number of Experiments	Average Score	Difference
First Test Series	8	2.6	0.1
Optimized	8	2.7	
Data Input			
Q-BOM	8	2.4	0.5
A-BOM	8	2.9	
Sequence	8	2.7	0.1
Without Sequence	8	2.6	
T-t-C	8	2.8	0.3
Without T-t-C	8	2.5	
Best Experiments			
First study (Meyer et al.)	1	3.1	0.2
This study	1	3.3	

As shown in Figure 5, the best result was achieved in the 16th experiment with an average score of 3.3. In this experiment, an optimized prompt chaining method was used, and information was provided using an Assembly BOM, Assembly Sequence, and Tools-to-Component relations. While this improved the information quality by 0.2 points compared to the best result from the initial study by Meyer et al. (2024), there is still a shortfall of 0.7 points to reach a fully satisfactory assembly instruction (a score of 4 across all criteria and elements of the instruction).

DISCUSSION

Building on the presented results, the subsequent section analyses the findings in relation to the initial hypotheses. The outcomes are examined to determine the extent to which the hypotheses are confirmed and to explore potential implications for future research and practical applications.

Prompt Chaining Comparison: The hypothesis that an optimized prompt chaining method with reduced information leads to higher information quality in assembly instructions can only be partially confirmed. The difference between the optimized prompt chaining (ø 2.7 points) and the prompt chaining that provides more information (ø 2.6 points) resulted in only a 0.1-point improvement. However, studies like those by Wei et al. (2022) and Reynolds and McDonell (2021) support the idea that focused, less complex prompts can enhance model performance.

Bills of Materials: In contrast to the study by Meyer et al. (2024), this study shows that structured data, such as those found in an Assembly BOM, are significantly better understood by the RAG model, leading to higher information quality in the assembly instructions. While using a Quantity BOM results in an average score of 2.4, the information quality improves to 2.9 with the use of an Assembly BOM, an increase of 0.5 points. Therefore, the hypothesis that utilizing RAG and ChatGPT-4o enables more effective processing of structured data from an Assembly BOM can be confirmed.

Assembly Sequences: The hypothesis that providing assembly sequences – i.e., information about the order of assembly steps – leads to higher information quality in the assembly instructions can also be confirmed. Including assembly sequences contributes to better structuring of the instructions and resulted in improved information quality, particularly in terms of how the information is organized. This finding is supported by the work of Wang et al. (2023), who emphasize that clearly structured data are crucial for the performance of LLMs.

Tools: The hypothesis that providing Tools-to-Component relations positively impacts the information quality of the assembly instructions can also be confirmed. Including these relations increased the information quality by an average of 0.3 points compared to instructions where this information was not provided. This result demonstrates that clearly defined relationships between tools and components improve the quality of the generated instructions.

Best Results: The hypothesis that the best result of this study surpasses the best result from the initial study by Meyer et al. (2024) can be confirmed. While the best result in this study achieved an average score of 3.3, the best result in the first study was 3.1. Although the improvement of 0.2 points may seem small, it can be concluded that the methods and optimizations used in this study led to a higher overall information quality.

Critical Appraisal: To ensure the most objective assessment of information quality, the three components of the assembly instructions (Tool Notes, Assembly Notes, and Quality Notes) were evaluated separately using six criteria based on a Likert scale. Evaluation examples were also provided as guidance. However, it's important to acknowledge that, despite the effort to maintain objectivity, a slight influence from personal preferences cannot be entirely ruled out. Additionally, the use of a four-point scale offers only limited differentiation in the evaluation.

CONCLUSION

The experimental results demonstrate that the RAG approach has potential for the automatic generation of assembly instructions. A key success factor has been the use of RAG and ChatGPT-4o, combined with the provision of structured datasets, such as Assembly BOMs and defined Tools-to-Component relations. Although the information quality of the assembly instructions was slightly improved compared to the initial study by Meyer et al. (2024), it is still insufficient. As a result, in practical applications, each generated instruction would need to be reviewed and refined by an employee.

Future research should not only focus on advancing the RAG methodology and optimizing data structures but also on exploring the implementation of feedback loops. For example, assembly workers (or participants in experimental trials) could evaluate the automatically generated instructions and provide feedback to the RAG system, enabling a continuous improvement process. These feedback loops would allow the creation of assembly instructions to be continuously monitored and refined, enhancing

the reliability of the instructions and ensuring higher quality outcomes in the long term.

ACKNOWLEDGEMENT

This research paper in the project KIPRO is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr which we gratefully acknowledge. D.tec.bw is funded by the European Union – NextGenerationEU.

REFERENCES

Bahr, L., Wehner, C., Wewerka, J., Bittencourt, J., Schmid, U., Daub, R. (2024) Knowledge Graph Enhanced Retrieval-Augmented Generation for Failure Mode and Effects Analysis; SSRN Elsevier. doi: https://dx.doi.org/10.2139/ssrn.4965185.

Buehler, M. J. (2024). Generative Retrieval-Augmented Ontologic Graph and Multiagent Strategies for Interpretive Large Language Model-Based Materials Design, American Chemical Society Engineering Au, Volume 4 No. 2, pp. 241–277, doi: https://doi.org/10.1021/acsengineeringau.3c00058.

Chandrasekhar, A., Chan, J., Ogoke, F., Ajenifujah, O., Farimani, A. B., (2024). AMGPT: A large language model for contextual querying in additive manufacturing, Additive Manufacturing Letters, Volume 11, doi: https://doi.org/10.1016/j.addlet.2024.100232.

Freire, S. K., Foosherian, M., Wang, C., Niforatos, E. (2023). Harnessing Large Language Models for Cognitive Assistants in Factories, 5th International Conference on Conversational User Interfaces (CUI), Eindhoven, Netherlands, 2023 doi: https://doi.org/10.1145/3571884.3604313.

Freire, S. K., Wang, C., Foosherian. M., Wellsandt, S., Ruiz-Arenas, S., Niforatos, E. (2024). Knowledge sharing in manufacturing using LLM-powered tools: user study and model benchmarking, Frontiers in Artificial Intelligence, Volume 7, doi: https://doi.org/10.3389/frai.2024.1293084.

Gao, Y., Xiong, Y,. Gao, X., Jia, K., Pan, J., Bi, X., Dai, Y., Sun, J., Wang, M., Wang, H. (2023) Retrieval-Augmented Generation for Large Language Models: A Survey, arXiv, doi: https://doi.org/10.48550/arXiv.2312.10997.

Hvam, L., Hansen, C. L., Forza, C., Mortensen, N. H., Haug, A. (2020). The reduction of product and process complexity based on the quantification of product complexity costs, International Journal of Production Research, Volume 58 No. 2, pp. 350–366. doi: https://doi.org/10.1080/00207543.2019.1587188.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks, 34th International Conference on Neural Information Processing Systems (NIPS), Vancouver, British Columbia, Canada, 2020, doi: https://dl.acm.org/doi/abs/10.5555/3495724.3496517.

Machado, D. M. (2024). Development of a cognitive assistant for industrial maintenance based on retrieval-augmented generation (RAG) at STMicroelectronics, Repositorio Institucional da Universidale Federal de Santa Catarina (RIUFSC), doi: https://repositorio.ufsc.br/handle/123456789/256433.

Meyer, F., Freitag, L., Hinrichsen, S., Niggemann, O. (2024). Potentials of Large Language Models for Generating Assembly Instructions, 29th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Padova, Italy, 2024, doi: https://doi.org/10.1109/ETFA61755.2024.10710806.

Reynolds, L., McDonell, K. (2021). Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm, Conference on Human Factors in Computing Systems (CHI), Yokohama, Japan, 2021, doi: https://doi.org/10.48550/arXiv.2102.07350.

Schuh, G., Rudolf, S., Riesener, M., Dölle, C., & Schloesser, S. (2017). Product production complexity research: Developments and opportunities, Procedia CIRP, Volume 60, pp. 344–349, doi: https://doi.org/10.1016/j.procir.2017.01.006.

Wang, Z., Zhong, W., Wang, Y., Zhu, Q., Mi, F., Wang, B., Shang, L., Jiang, X., Liu, Q. (2023) Data Management for Large Language Models: A survey, arXiv, doi: https://doi.org/10.48550/arXiv.2312.01700.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., Zhou, D. (2022). Chain-of-Thought prompting elicits reasoning in large language models, 36th International Conference on Neural Information Processing Systems (NIPS), New Orleans LA, USA, 2022

Xia, Y., Xiao, Z., Jazdi, N., Weyrich, M. (2024). Generation of Asset Administration Shell with Large Language Model Agents: Toward Semantic Interoperability in Digital Twins in the Context of Industry 4.0, IEEE Access, Volume 12, pp. 84863–84877, doi: https://doi.org/10.1109/ACCESS.2024.3415470.