Advancing Capability Matching in Manufacturing Reconfiguration With Large Language Models

Fan Mo[1,2], Jack C. Chaplin[1] David Sanderson[2], and Svetan Ratchev[1,2]

[1] Institute for Advanced Manufacturing, University of Nottingham, Nottingham, Nottinghamshire, NG8 1BB, United Kingdom
fan.mo@nottingham.ac.uk

[2] Centre for Aerospace Manufacturing, University of Nottingham, Nottinghamshire, NG7 2GX, United Kingdom

Abstract This paper introduces an approach that integrates Natural Language Processing (NLP) and knowledge graphs with Reconfigurable Manufacturing Systems (RMS) to enhance flexibility and adaptability. We utilize a chatbot interface powered by GPT-4 and a structured knowledge base to simplify the complexities of manufacturing reconfiguration. This system not only boosts reconfiguration efficiency but also broadens accessibility to advanced manufacturing technologies. We demonstrate our methodology through an application in capability matching, showcasing how it facilitates the identification of assets for new product requirements. Our results indicate that this integrated solution offers a scalable and user-friendly approach to overcoming adaptability challenges in modern manufacturing environments.

Keywords: natural language processing, chatbot, large language model, manufacturing reconfiguration, knowledge graph

1 Introduction

Manufacturing is evolving due to increased demand for personalized products and resource unpredictability. This change requires agile, adaptable systems, leading to the adoption of Reconfigurable Manufacturing Systems (RMS) [1]. RMS combines the high output of Dedicated Manufacturing Lines (DML) [2] with the flexibility of Flexible Manufacturing Systems (FMS) [3], allowing for quick adaptation to market shifts while maintaining efficiency. However, reconfiguring manufacturing processes to meet varying demands presents challenges, as traditional methods often lack the required dynamism and intuitive interfaces for effective decision-making [4].

Addressing these challenges, recent advancements in Natural Language Processing (NLP) [5], knowledge graphs [6], and Large Language Model (LLM) [7] present a promising solution. The integration of these technologies into Reconfigurable Manufacturing System (RMS) can significantly enhance data interaction and system efficiency. By leveraging LLMs and knowledge graphs, this paper introduces a methodology that substantially improves the reconfiguration process in manufacturing systems. Central to our approach is a chatbot interface that enables seamless, intuitive user interactions, facilitating a more efficient, accessible, and user-friendly reconfiguration process. This marks a significant advancement in the field of manufacturing system design and optimization.

The paper is organized as follows: Section 2 reviews relevant literature on LLMs, knowledge graphs and chatbot. Section 3 details the proposed methodology, emphasizing the integration of NLP, LLMs, and knowledge graphs in RMS. Section 4 validates our approach through one case study in capability matching. Finally, Section 5 concludes the paper, highlighting our findings and future research directions.

2 Related work

2.1 Large Language Models and Generative Pre-trained Models

The field of artificial intelligence (AI) encompasses a wide array of technologies, among which LLMs [8] and Generative Pre-trained Models (GPTMs) [9] represent significant areas of innovation and application. These models have fundamentally transformed our approach to NLP, making it possible to interact with digital systems more intuitively and human-like.

LLMs are sophisticated AI systems designed to understand, generate, and interact with human language at a remarkable scale. These models are trained on extensive datasets comprising texts from diverse sources, enabling them to grasp the nuances of language, including grammar, context, and semantics. LLMs can perform a variety of tasks, such as text generation, summarization, translation, and more, making them invaluable tools in many domains [10].

2.2 Knowledge Graph

The knowledge graph concept represents a transformative approach to managing and utilizing information, highlighted by the introduction of Google's Knowledge Graph project in 2012 [15]. This initiative aimed to enhance the search engine's ability to deliver relevant results by understanding the relationships between different pieces of information, thus significantly improving user experience.

A knowledge graph is a structured form of knowledge representation in triples that connects entities (such as objects, events, or concepts) through various relationships [16]. This structure is particularly adept at representing heterogeneous data, allowing for the nuanced depiction of the complexity inherent in real-world information. The strength of knowledge graphs lies in their dual-layer architecture: the schema layer and the entity layer.

The schema layer provides the foundation for the knowledge graph, defining the types of entities that exist within the graph and the possible relationships between them. It establishes a formal structure for the data, ensuring consistency and enabling effective reasoning and inference. The schema layer adheres to standards like Resource Description Framework (RDF) and Web Ontology Language (OWL), set by the World Wide Web Consortium, which facilitate the creation of application-independent, knowledge-centric graphs that are understandable by both humans and machines.

3 Proposed methodology

Motivated by the scarcity of research in manufacturing reconfiguration for chatbots and the accuracy challenges due to data representation shortcomings, there's a critical need for innovative approaches. A methodology for constructing the chatbot using LLMs has been proposed. Its purpose is to facilitate interaction between humans and robots through speech or text. The chatbot's structure comprises the following components:

1. **Human-Machine Interplay Block:** This block facilitates user and machine communication. It is designed to present information to the user and gather user inputs, functioning across web browsers, desktop software, or graphical user interfaces (GUIs).
2. **Logic Block:** The core processing unit where the business logic resides, this block interprets the user's intent, accesses the necessary data, and determines the appropriate response based on the chatbot's algorithms and rules.
3. **Data Block:** Serving as the repository of all knowledge, this block manages and retrieves information from a structured Knowledge Base, ensuring that the chatbot's responses are informed and contextually relevant.

The relationship among the three components is illustrated in Figure 1, depicting a sequence that begins with the human-machine interplay system receiving a query from the user. This query is passed to the logic block, where natural language processing and logical analysis occur. Concurrently, the data block retrieves necessary information and provides support to the logic block, facilitating further processing if required. After processing the information, the logic block generates a response, which is then displayed to the user by the human-machine interplay system, completing the cycle of interaction.

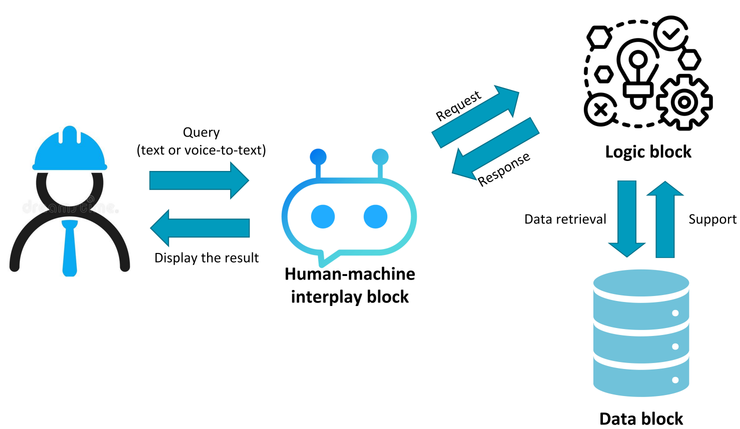

Figure 1: Workflow diagram depicting the interaction between the Human-Machine Interplay Block, Logic Block, and Data Block.

3.1 Human-Machine Interplay Block

The Human-Machine Interplay Block acts as the interface for direct interaction between the user and the chatbot, primarily facilitating the reception of user inputs and the presentation of the chatbot's responses. This block is tailored to ensure an intuitive and seamless communication flow, exclusively handling the input and output processes without delving into the complexities of input interpretation, which the Logic Block manages.

- **Receiving Input:** Captures and forwards user queries to the Logic Block. Input can be in the form of text or, if submitted as voice, is converted to text via Automatic Speech Recognition (ASR) technology.
- **Displaying Results:** Presents the processed responses from the chatbot to the user, utilizing a clear and user-friendly format.

3.2 Logic Block

The Logic Block serves as the central processing unit of our chatbot framework, intelligently orchestrating the flow of information from user inputs to generating appropriate responses. This block processes inputs received through the Human-Machine Interplay Block, utilizes the NLP techniques for comprehension and intent determination, and formulates responses based on a detailed Knowledge Base (KB). Within this block, sophisticated decision-making algorithms are employed to handle user queries by efficiently

- Parsing and comprehending user inputs through advanced NLP techniques.
- Retrieving relevant information from the KB.
- Utilizing public LLMs like GPT-3 or GPT-4 when the KB does not provide a sufficiently accurate answer, ensuring a broader scope of response generation.
- Employing decision-making algorithms to select the optimal response, prioritizing answers from the KB for precision and supplementing with LLM-generated responses when necessary.

3.3 Data Block

The Data Block's effectiveness in our chatbot framework is significantly amplified through the integration of a knowledge graph designed to represent and manage heterogeneous data accurately while facilitating machine learning applications. The knowledge graph is structured into two principal layers, each serving a distinct purpose in the data management ecosystem:

- **Schema Layer:** Defines the architecture of the graph, specifying entity types and their possible interrelations.
- **Entity Layer:** Contains the instantiated data that populates the schema structure.

The knowledge graph's construction from diverse information sources is guided by the need to support complex reasoning and facilitate machine learning. Through these mechanisms, the Data Block ensures the chatbot's responses are precise, relevant, and reflective of the latest available information, thereby enhancing the overall user experience.

4 Implementation

4.1 Construction of the ChatBot

In alignment with the proposed methodology outlined in Section 3, the construction of our chatbot encompasses three essential components. These include the design of the human-machine interplay block, the logic block, and the data block.

The human-machine interplay block is responsible for visualizing the interaction between humans and machines. We have implemented this block using the React Simple Chatbot package as the visualization interface, which is integrated into our system with JavaScript. This package was selected for its robustness, ease of use, and compatibility with modern web technologies. The implementation of this block is a critical step in ensuring that our chatbot is accessible and user-friendly, providing a solid foundation for the sophisticated processing capabilities that follow in the subsequent blocks.

In the Logic Block of our chatbot framework, we utilize advanced NLP techniques and harness the capabilities of public LLMs like GPT-3 or GPT-4, while also leveraging a structured KB for domain-specific accuracy. This setup allows us to evaluate the sufficiency of the KB's response for a given query. If the response from the KB is deemed adequate, it is used for its precise domain-specific information. Otherwise, the query is directed towards an LLM for a broader understanding. This approach ensures a versatile and adaptive response mechanism capable of addressing a wide array of user queries with high accuracy and relevancy. To facilitate this, we have integrated the OpenAI API into our system using JavaScript, enabling seamless interaction between the logic block and the human-machine interplay block. This integration not only showcases the efficient use of GPT-4 for generating responses but also highlights the system's scalability and adaptability, significantly enhancing the user experience by providing timely and contextually relevant information.

The Data Block is a critical component of our chatbot framework, primarily focused on creating, managing, and retrieving knowledge from the KB. It encompasses the comprehensive development of a knowledge graph and integrates closely with the Logic Block for efficient data retrieval based on user queries. This section is designed to leverage KB creation information detailing its construction, integration, and operational use within the chatbot.

The development of the KB is achieved by constructing a knowledge graph specifically tailored for the domain of manufacturing reconfiguration, following the methodology outlined by Mo et al. [17]. This approach results in a complex structure adept at encapsulating the diverse and intricate nature of data within manufacturing.

The Data Block's capabilities, critical for the chatbot's operation, extend into advanced data retrieval mechanisms intricately integrated with the Logic Block. This integration is pivotal for efficiently processing user queries and retrieving pertinent data from the KB to generate responses. To facilitate this sophisticated interaction between the database and the Logic Block, we employ the Neo4j-Driver API for JavaScript. This choice of technology enhances our retrieval process as follows:

- **Neo4j-Driver API Integration:** Through the Neo4j-Driver API, custom algorithms within the Logic Block directly interact with the graph database, executing Cypher queries to fetch relevant data. This direct integration ensures seamless communication between our chatbot's processing logic and the KB stored in Neo4j, enabling efficient and accurate data retrieval. Neo4j provides an accurate answer to solve the limitation of the public LLM.

- **Query Interpretation:** The Logic Block, utilizing GPT-4's NLP capabilities, interprets user queries and translates them into specific, actionable Cypher queries for the knowledge graph. This translation is crucial for making the retrieval requests precise and tailored to the user's needs. If the answer from the Neo4j is not available, then this one will used to generate a general answer.
- **Data to Response Conversion:** Once data is retrieved, the Logic Block transforms this information into coherent and informative responses. Leveraging the response generation capabilities of GPT-4, the chatbot articulates the data into user-friendly answers, providing insights and information directly derived from the KB.

4.2 Application in Capability Matching

Capability matching emerges as a critical function within the context of adapting manufacturing systems to new product requirements, often driven by specific customer demands. This adaptive process necessitates a real-time evaluation of the manufacturing system's capabilities in relation to the new product specifications. Our chatbot system plays a crucial role in facilitating this evaluation, leveraging advanced NLP and database query technologies. A typical user interaction scenario might involve querying the system about the assets required for a specific task, such as "What are the required assets for task 100?"

Initially, the user submits a query related to manufacturing reconfiguration, specifically targeting capability matching for a new task. This query is input into the chatbot interface, where the GPT-4 model processes the natural language input, translating it into Cypher query language understandable by the Neo4j database. Upon execution of this query, the database returns a list of candidate assets suitable for the task at hand.

For instance, as depicted in the process flow, querying for the execution requirements of task 100 yields a selection of candidate assets, including "Fanuc M-900ib/400L", "KUKA KR 240 R3330", "Fanuc M-2000iA/1700L", among others. Subsequent inquiries about specific assets, such as requesting the cost of "Fanuc M-900iB/400L", are similarly processed by the chatbot, with the response generated based on data retrieved from the Neo4j database, as shown in Figure 2.

This integrated chatbot system, by converting user inquiries into actionable database queries and presenting the information in a user-friendly manner, significantly enhances the decision-making process in manufacturing reconfiguration. It exemplifies the practical application of combining AI-driven NLP with graph database technologies to streamline complex operations within industrial settings.

5 Conclusions

This paper presented an approach leveraging advancements in LLMs and knowledge graphs to address the complexities of reconfiguring manufacturing systems. By integrating NLP capabilities with a robust knowledge representation framework, we proposed

Figure 2: Displaying cost information from the chatbot

a methodology that significantly enhances the adaptability and efficiency of RMS. Our approach democratizes access to RMS, making it more accessible to users with varying levels of technical expertise, thereby facilitating a broader adoption of this technology in the manufacturing industry.

Throughout this study, we demonstrated the application of our methodology in one critical manufacturing reconfiguration area: capability matching. It highlighted the practical benefits of our approach, showcasing its potential to streamline the reconfiguration process and improve operational efficiency. By enabling more effective interaction between users and manufacturing systems, our methodology offers a path toward more agile and responsive manufacturing operations that adapt to changing market demands and customer preferences.

References

1. Mo, F., Chaplin, J.C., Sanderson, D., Rehman, H.U., Monetti, F.M., Maffei, A., Ratchev, S.: A Framework for Manufacturing System Reconfiguration Based on Artificial Intelligence and Digital Twin. Proceedings of the International Conference on Flexible Automation and Intelligent Manufacturing, 361–373 (2022). Springer.
2. Adenso-Diaz, B., Lozano, S.: A Model for the Design of Dedicated Manufacturing Cells. International Journal of Production Research 46(2), 301–319 (2008). doi:10.1080/00207540600942235
3. Yadav, A., Jayswal, S.C.: Modelling of Flexible Manufacturing System: a Review. International Journal of Production Research 56(7), 2464–2487 (2018). doi:10.1080/00207543.2018.1449978

4. Mehrabi, M.G., Ulsoy, A.G., Koren, Y.: Reconfigurable Manufacturing Systems: Key to Future Manufacturing. Journal of Intelligent Manufacturing 11, 403–419 (2000). doi:10.1023/A:1008980401450
5. Chowdhary, K.R.: Natural Language Processing. Fundamentals of Artificial Intelligence, 603–649 (2020). Springer.
6. Martínez-Arellano, G., Niewiadomski, K., Mo, F., Elshafei, B., Chaplin, J.C., McFarlane, D., Ratchev, S.: Enabling Coordinated Elastic Responses of Manufacturing Systems through Semantic Modelling. IFAC-PapersOnLine 56(2), 7402–7407 (2023). Elsevier.
7. Hadi, M.U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M.B., Akhtar, N., Wu, J., Mirjalili, S., et al.: A Survey on Large Language Models: Applications, Challenges, Limitations, and Practical Usage. Authorea Preprints (2023).
8. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al.: A Survey on Evaluation of Large Language Models. ACM Transactions on Intelligent Systems and Technology (2023).
9. Jin, Z., Jiang, X., Wang, X., Liu, Q., Wang, Y., Ren, X., Qu, H.: Numgpt: Improving Numeracy Ability of Generative Pre-Trained Models. arXiv preprint arXiv:2109.03137 (2021).
10. Xiao, C., Xu, S.X., Zhang, K., Wang, Y., Xia, L.: Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase of ChatGPT in Education Applications. Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), 610–625 (2023).
11. Firat, M.: How Chat GPT Can Transform Autodidactic Experiences and Open Education. Department of Distance Education, Open Education Faculty, Anadolu University (2023).
12. Chen, K., Cosgro, B., Domfeh, O., Stern, A., Korkmaz, G., Kattampallil, N.A.: Leveraging Google BERT to Detect and Measure Innovation Discussed in News Articles. 2021 Systems and Information Engineering Design Symposium (SIEDS), 1–6 (2021). IEEE.
13. Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., Liu, Z., Prabhumoye, S., Zerveas, G., Korthikanti, V., et al.: Using Deepspeed and Megatron to Train Megatron-turing NLG 530B, a Large-Scale Generative Language Model. arXiv preprint arXiv:2201.11990 (2022).
14. Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., Kashinkunti, P., Bernauer, J., Catanzaro, B., et al.: Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 1–15 (2021).
15. Fensel, D., Şimşek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Toma, I., Umbrich, J., Wahler, A., Fensel, D., et al.: Introduction: What is a Knowledge Graph? Knowledge Graphs: Methodology, Tools and Selected Use Cases, 1–10 (2020). Springer.
16. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Kirrane, S., Labra Gayo, J.E., Navigli, R., Neumaier, S., et al.: Knowledge Graphs. Synthesis Lectures on Data, Semantics, and Knowledge 12(2), 1–257 (2021). Morgan & Claypool Publishers.
17. Mo, F., Chaplin, J.C., Sanderson, D., Martínez-Arellano, G., Ratchev, S.: Semantic Models and Knowledge Graphs as Manufacturing System Reconfiguration Enablers. Robotics and Computer-Integrated Manufacturing 86, 102625 (2024). doi:10.1016/j.rcim.2023.102625