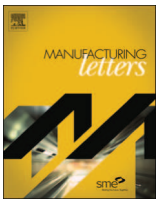
Available online at www.sciencedirect.com

Manufacturing Letters

Manufacturing Letters 41 (2024) 253–263



52nd SME North American Manufacturing Research Conference (NAMRC 52, 2024)

Framework for LLM Applications in Manufacturing

Cristian I. Garcia[a,*], Marcus A. DiBattista[a], Tomás A. Letelier[a], Hunter D. Halloran[a], Jaime A. Camelio[a]

[a]*Innovation Factory, College of Engineering, University of Georgia, 302 E Campus Rd., Athens, GA 30602 United States*

Abstract

In the era of Industry 4.0, the proliferation of data within manufacturing environments has presented both unprecedented opportunities and challenges. This paper introduces a framework that capitalizes on the capabilities of Large Language Models (LLMs) to revolutionize data integration and decision-making processes in manufacturing systems. Addressing the critical need for efficient data management, our framework streamlines the consolidation, processing, and generation of responses to essential inquiries, thus enhancing manufacturers' capabilities to extract valuable insights. The focus of this paper is twofold. First to establish a framework for the use of LLM applications in manufacturing settings. Secondly, to provide an overview of the manufacturing connection between data, AI, and chat-bots, while also addressing a few pain points identified from the manufacturing literature. The paper then introduces FILLIS (*Factory Integrated Logic and Language Interface System*), a Large Language Model assistant, through a compelling case study. FILLIS showcases remarkable versatility, excelling in tasks ranging from elucidating machine operations to language translation. The study underscores FILLIS's proficiency in handling specific contexts, answering questions from uploaded documents with precision. However, inherent limitations surface in tasks involving mathematical operations, emphasizing the need for external agents in specific scenarios. This pivotal opportunity is explored in the proposed framework as it advocates for integrating external agents alongside LLMs, creating a more versatile and comprehensive assistant tool. The findings of this paper and proposed framework position LLMs as transformative tools for intelligent data processing.

Keywords: Large Language Models; Chat-bot; Natural Language Processing; Big Data; Human-Computer Interaction; Embeddings; IoT; Industry 4.0;

1. Introduction

Due to the recent Industry 4.0 explosion, Internet of Things (IoT) applications have made their way into increasingly more manufacturing settings, enabling the centralized acquisition of data that was often spread all over not just several departments, but across the entire facility. With simultaneous access to all this information originating from different parts of a manufacturing plant, there's potential to generate insight that was previously extremely difficult to obtain. However, there are still some major challenges when it comes to handling this new sea of information.

This newly available data requires a considerable degree of multidisciplinary expertise each sector involved to be able to interpret it all; discern between relevant and redundant information; and, even more importantly, be able to correlate and extrapolate any kind of conclusion from it. In the face of these enormous data volumes, leveraging Artificial Intelligence (AI) applications is not just convenient, but essential. AI enables efficient processing, analysis, and extraction of valuable insights from vast and complex datasets, enhancing decision-making and productivity across diverse domains.

In recent years, AI applications have also found homes in many areas of industry; manufacturing is no exception. There has been increased interest in Machine Learning (ML) applications, especially in the academic community, as evidenced by the ever-increasing publication count [1]. Despite growing popularity, most branches of AI are still far from saturated, as

* Corresponding author. E-mail address: cristian.garciaponce@uga.edu

is especially the case with Natural Language Processing.

Natural Language Processing (NLP) is one of the many branches of AI, but this one focuses on working with written language. NLP models seek to better enable communication between human and machine, by attempting to create digital models capable of understanding sources of information in text form, as well as being able to produce human readable responses from data. The popularity of NLP has increased drastically in recent years, which has led to a surge in research on NLP and Large Language models. Despite this, there are still a small number of papers regarding the use of NLP applications in the manufacturing sector; most of the literature has been more focused on the theoretical development of NLP algorithms and models [2].

Large Language Models (LLMs) are a segment of NLP models, consisting of deep neural networks capable of both understanding and generating human-like language in written form. LLMs have gathered a lot of attention recently, largely due to open models like LLAMA and services like ChatGPT making both hobbyists and academics start thinking of ways to apply this technology into different fields. However, as mentioned before, there is still much to explore regarding applications of LLMs in manufacturing [2]. LLMs are capable of both understanding different forms of data and information, as well as adapting gathered knowledge into different contexts. Being able to format the presentation of data based on whether the user is a chief engineer or sales manager enables a greater and more streamlined understanding of the information available and allows for faster and enhanced decision making.

There exist many manifestations of NLP models, from more simplified and straightforward chat-bot NLP architectures [3] to more complex and complete models [4], as well as plenty of papers and documentation on how to properly preprocess data to utilize it in an NLP model [5]. However, not many delve into the question of what kinds of data to use in these models, and thus it remains unclear what data should or could be used in a manufacturing setting, as well as what kind of information can and should be expected from the results.

The goal of this paper is to establish some guidelines for the use of LLM applications in manufacturing settings. In the next section we first provide an overview of the manufacturing connection between data, AI, and chat-bots, while also addressing a few pain points identified from the manufacturing literature. As LLMs can't be expected to excel in every scenario, we present a few key characteristics, functions, and design dimensions of tasks that are better suited for these models to help identify use cases that would be more fitting for this kind of tools. Afterwards, we propose a framework that outlines the general structure of an LLM assistant with the intention of bringing clarity to both the structure and functionality of such applications. We conclude by implementing the suggested guidelines in a case study involving the development of a virtual assistant designed to perform diverse tasks in support of manufacturing operations.

2. Related Work

2.1. Data in manufacturing

The significance of data in manufacturing is recognized by its ability to empower information-driven decisions which enhances a manufacturer's competitiveness. This is accomplished by utilizing patterns and trends in the data to identify certain characteristics, such as areas for improvement. This ability to signal key features allows manufacturers to gain vital insights into their processes [6]. Furthermore, data in manufacturing is significant in optimizing production schedules, enhancing supply chain management, and facilitating the integration of smart manufacturing technologies. These capabilities represent key aspects of the escalating emphasis on data within the realms of manufacturing and production research [6, 7]. This enduring focus on data lays the foundation to advance manufacturing capabilities, aligning with the evolving landscape of smart manufacturing where technologies such as artificial intelligence and machine learning play an increasingly crucial role.

2.2. LLMs in manufacturing

In the context of Industry 4.0, the emergence of the Internet of Things (IoT) and Artificial Intelligence (AI), particularly Machine Learning (ML), offers transformative solutions for manufacturers in processing and leveraging their data efficiently [8]. The extensive volume of data generated in modern manufacturing, often referred to as Big Data, presents both challenges and opportunities. AI tools present themselves as one of the opportunities when applied to manufacturing data because they enable functionalities such as planning, teaching, operating, monitoring, intervening, and learning [9]. These functionalities are accomplished by developing algorithms and models that can process the data with varying degrees of autonomy and scale. These models, which are referred to as AI, are designed to perform specialized tasks. The type of models this study will focus on are Large Language Models (LLMs), which as mentioned earlier are capable of understanding and generating human-like language in written form. However, the increasing complexity of AI and its "black box" outputs, specifically models utilizing unsupervised training known as deep learning models, poses a challenge for non-specialists in comprehending and interpreting the results accurately. Bridging this knowledge gap is crucial for facilitating informed decision-making, prompting research into explainable AI models (XAI) that aim to make AI outputs more accessible to those without specialized expertise [9]. Additionally, the gap provides an opportunity for LLMs as they can provide code, point to resources, or conversationally walk through the logic of ideas. Therefore, while the accessibility of these AI tools will continue to improve, by demonstrating to manufacturers how AI can help them achieve desired results such as high equipment effectiveness and a re-

silient manufacturing system, this will guarantee AI and ML emerges as a vital tool [10].

2.3. Chat-bots in manufacturing

This growing theme of AI in manufacturing continues with the integration of LLMs as conversational agents, particularly chat-bots, as an interface between humans and their manufacturing environment [11]. Recent work builds on this by providing insights into the necessary elements for developing conversational agents in manufacturing, emphasizing technical characteristics and functional aspects [12]. It acknowledges the scarcity of real-world manufacturing case studies involving conversational agents, emphasizing their configuration as smart solutions applicable to various processes [12]. The research highlights chat-bots' intervention in both repetitive and hazardous operations, particularly their potential impact on operational efficiency and operator safety [12]. Additionally, it provides an important foundation for understanding the potential of LLMs to improve other AI models' performance; however, the framework presented failed to exploit LLMs beyond the conversational capabilities. We will delve further into the additional capabilities of LLMs in later sections. Meanwhile, to further elaborate on the significance of practical LLM applications, additional studies have organized the critical information essential for future research in the chat-bot field by providing a historical evolution and architectural design considerations [13]. This includes summarizing the evolving landscape of AI-based chat-bots through co-citation analysis, offering insights into thematic content, leading authors, and contributing institutes [13, 14]. Collectively, these studies lay the groundwork for exploring the integration of chat-bot applications into the manufacturing environment, emphasizing the need for tailored solutions and further empirical validation through real-world case studies.

Recent research has begun to explore the novel applications of embeddings and their role as vector representations capturing semantic and contextual information. This will be explored in greater detail in the framework section, however, functionally these embeddings allow LLMs to enable more capabilities beyond just being conversational agents. A recent study focused on creating a personalized and responsive LLM-powered customer service agent used embeddings to be able to produce a chat-bot assistant that significantly improved their customer-company relationship beyond previous measures [15]. Another study focused on how embeddings and a LLM Model can collaboratively power a PDF chat-bot, demonstrated the synergy of both technologies in creating scalable AI/LLM applications [16, 17]. While these studies are few and limited, this emerging body of work demonstrates the versatility of embeddings in various applications. As the manufacturing industry continues to embrace data-driven technologies, these advancements present promising avenues for efficiency, responsiveness, and innovation.

Shifting from the research done in LLM assistant architectures and chat-bot applications using embeddings to current examples of chat-bots being used, we find several examples of them playing pivotal roles. In one study, the researchers focused on applications to enhance operational efficiency. Specifically, within the procurement phase of the supply chain, chat-bots were integrated with Enterprise Resource Planning (ERP) systems to strategically add value by automating the supplier selection processes which led to minimized losses due to failed supplier relations [18]. This study demonstrated the positive impact of digitalization on supply chain metrics, presenting opportunities for the efficient utilization of chat-bots, despite a prevalent lack of Robotic Process Automation (RPA) adoption. Moreover, in domains like maritime transportation, researchers developed a chat-bot called Popeye' to improve employee health and safety by training employees in safety-critical operations when dealing with sea container terminals. Popeye's real-time interaction aided in procedure adherence which led to increased training efficacy and reduced cognitive load, underscoring the value of deployable chat-bot applications [4]. These few examples emphasize the broad utility of LLM assistants/chat-bots in diverse contexts and provide an opportunity for their applications in manufacturing environments to be explored.

2.4. Current pain points in manufacturing

Several research endeavors have shed light on the challenges faced by manufacturers, particularly Small and Medium-sized Enterprises (SMEs), as they strive to adopt and implement advanced manufacturing paradigms such as Lean Manufacturing and Industry 4.0. The results of research performed emphasizes the critical role of training, managerial involvement, and external expert support in ensuring the success of Industry 4.0 projects within SMEs [19]. Additionally, the identified risks and opportunities in Industry 4.0 adoption further highlight the need for targeted strategies and comprehensive support mechanisms [19]. This provides an interesting avenue for LLMs as one of the capabilities they possess is the high domain knowledge of the material they have been trained in. Additionally, research has been done which expands on the synergy between lean manufacturing and factory digitalization, which delves into the relationships among these practices and their collective impact on operational performance [20]. The findings underscore the importance of embracing both Lean Manufacturing and digital technologies, with a synergistic effect noted when these strategies are combined. This further demonstrated the benefit that LLM assistants could have not only on the cyber-physical and data processing side but also on a manufacturer's total quality management (TQM) and manufacturing system paradigm. Which is further emphasized by recent work that compiles and identifies the gaps in current research on Industry 4.0 applied to SMEs [21]. The study reveals that while SMEs often limit themselves to adopting Cloud Computing and the Internet of Things, there remains a notable absence of real applications in production planning. This highlights a crucial void in the practical application of Industry 4.0 concepts within SMEs, signaling the necessity

for tailored solutions and more comprehensive frameworks [21]. Additional work has been done to investigate the critical success factors and their progression dependency for SMEs implementing lean manufacturing [22]. The study offers valuable insights into the dynamic nature of success factors, indicating that in the initial stages, SMEs can enhance their lean practices through local factors including learning focus and improvement training. As the practices advance, company-wide factors such as top management support become indispensable, emphasizing the need for a nuanced and progressive approach to lean implementation [22]. Returning to previous statements, LLMs possess capabilities to have both domain specialized knowledge and to provide tailored responses. The pain points identified by these studies present strong opportunities for LLM applications in manufacturing systems to be explored.

Lastly, a critical review of existing Smart Manufacturing (SM) and Industry 4.0 maturity models highlights a significant gap in addressing the specific requirements of SMEs [23]. The recent study advocates for the development of a realistic SM maturity model customized for SMEs, acknowledging the unique starting conditions, the disconnect between maturity models and self-assessment tools, and the need for tailored support beyond the assessment phase. Together, these studies contribute to a comprehensive understanding of the challenges manufacturers face in adopting transformative manufacturing paradigms, paving the way for innovative solutions such as LLM applications with specialized domain knowledge to assist and guide manufacturers through these complex processes.

Upon reviewing the relevant literature, three primary gaps are found that should be addressed in this study:

(1) Limited work has been done to explore how LLMs can help manufacturers navigate the lack of specialized expertise in their manufacturing system.

(2) Although there are tools available to process and evaluate data, there are no studies available that focus on using LLMs as a data processing system that integrates with a manufacturing system.

(3) The studies which attempt to bridge the theoretical knowledge and practical knowledge of LLM applications and chat-bots lack an operational and practical framework that would lead to a comprehensive LLM application.

3. Use Case Determination

As we transition from the foundational understanding of previously mentioned technologies and their connections to manufacturing, the focus now shifts towards practical applications of chat-bots, or as we will refer to them, LLM data assistants, within a manufacturing system. Think of the LLM data assistant as an intern: capable of handling tasks ranging from simple to moderately complex, provided it has access to the necessary information. While it might not be delivering morning coffee, this assistant excels at repetitive, time-consuming, and data retrieval tasks. To maximize its utility, establishing a standard for task delegation is crucial. Table 1 outlines optimal characteristics, functions, and design dimensions for tasks suitable for an LLM data assistant, facilitating the identification of tasks ripe for delegation. For instance, posing specific questions about equipment features aligns well with the assistant's strengths, requiring technical communication, domain-specific knowledge, and contextual referencing. This type of task not only leverages the model's capabilities but also underscores its ability to reference closed-domain information, offer support for users, and demonstrate proficiency in retrieving and generating instructions from equipment manuals. The following few sections will delve into the practical considerations and criteria guiding the identification of tasks where an LLM data assistant can be most impactful.

Table 1. Optimal Characteristics, Functions, and Design Dimensions for LLM Assistant Task Delegation.

Task Characteristics	LLM Functionality	LLM Design Dimension
Knowledge Domain	Intent Classification	History / Context Referencing
	Entity Identification	Domain Specific Knowledge
	Closed or Open Domain Referencing	Continuity Focused
Goal Oriented	Action Execution	Iterative
		Deliverable Focused
		Time Sensitivity
Service Provider	Data Processing	Boolean Inferencing
	Data Handler	Planning & Task Managing
	Retrieval Based	Data Handling
		Data Visualization
		Translation
User Focused	User Support	Personalized Interface
	Generative Based	Technical Communication
		Non-Technical Communication

4. Framework

The presented framework offers methodologies for integrating data across manufacturing systems and databases, a resource that is presently underexplored. This integration aims to leverage Large Language Models (LLMs) to streamline the consolidation, processing, and generation of responses to crucial inquiries concerning the entire manufacturing system. Before delving into the details of the framework, it is imperative to grasp the concept of embeddings and their role in the functioning of LLMs.

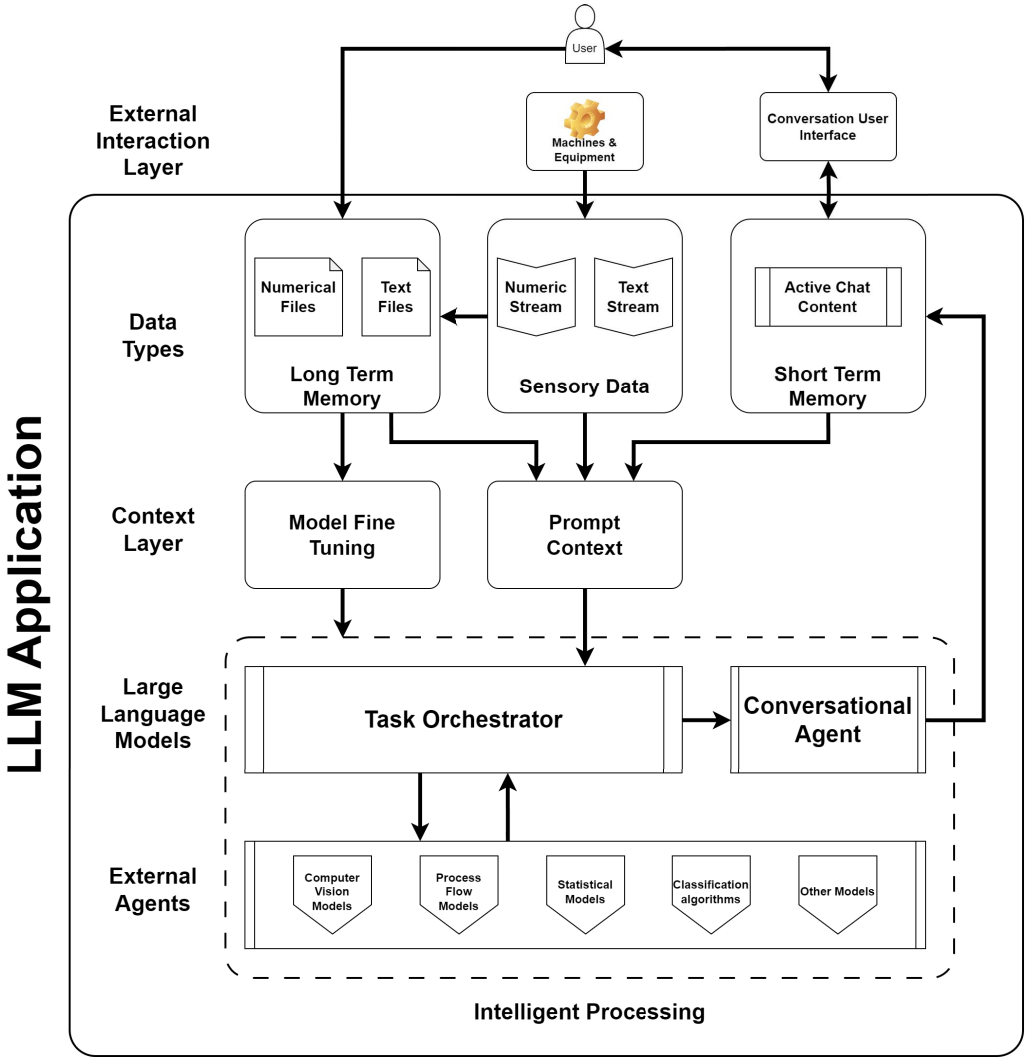


Figure 1. Proposed LLM Application Framework

4.1. Embeddings

Embeddings are the key element for LLM applications to understand written text. LLMs, as almost any machine learning model, work by using the "features" of an element, which numerically describe the characteristics of said element, as inputs. Much like features, embeddings consist of vectors that numerically represent the contents of text in a manner that a machine can understand. They essentially map words, sentences, and text to an N^{th} dimensional space [24]. This makes it possible to calculate correlation and similarity between texts through the "distance" from each other in this mapped space. To create these embedding vectors from text is not a trivial task, and because of that there exist many pre-trained deep learning models whose purpose is specifically to generate these embeddings from a given text sample.

Since embeddings are used to calculate correlation and similarity, it is important to be able to query these metrics on large databases in an efficient manner. Vector databases have been developed for this purpose, specifically designed to store embedding vectors along with references to their corresponding text files. In contrast to conventional databases that retrieve files based on exact query matches, vector databases are optimized to retrieve the most similar results from a query. This optimization aligns with the objective of LLMs to generate responses that are probabilistically correct. When retrieving information from a vector database, the query uses the input's text embeddings and compares it to the contents of the database, retrieving the text files whose corresponding embedding are the most similar. LLMs take advantage of this to quickly obtain relevant information from the database and use it when given a prompt, which produces an answer with accurate and relevant information.

4.2. External Interaction

Direct interaction with the model should primarily be done through a Conversation User Interface, which would consist of the "front-end" of the LLM application. Within this interface a

user should be able to request information to the model through text prompts, be it either directly written by the user, obtained from speech to text, selected among predefined options, or any other form of obtaining a written question. After a prompt, it is also through this interface that the user will receive it's answer from the LLM application, at which point the user should be allowed to prompt the model with a new question should they need to. While not mandatory for the model to work, it is ideal that the Conversation User Interface also displays the interaction history in some way, showing the latest prompts and corresponding answers, as these allow the User to generate more precise and detailed questions.

For the model to work, it is necessary to introduce information into its system. Either Users or Administrators should work on providing the model's long-term memory (described later) with any relevant historical data, which the model will use to generate its answers. If possible, it is also useful to link the model with any data being generated in real time, such as data from sales, machinery, or any other equipment which can constantly feed and update the model's memory and prompt context.

4.3. Data Types Layer

When interacting with an LLM, the data it's aware of and utilizes can be divided into Long-term memory, Short-term memory, and Sensory data. Long-term memory is composed of all types of information stored in a long-term manner, like the contents of a database or another storage unit, and is made accessible to the LLM. This data can be either numerical data, (such as machine sensor history, sales data, historical logs, etc.) text files (Machine or tool documentation, product catalogs, sales contracts, etc.), or a combination of both. These files and data streams often originate from various departments within an enterprise, which may not regularly interact yet may possess information pertinent to one another. Table 2 lists a few examples of files and data streams that enterprises may have and the respective department that owns them. Allowing an LLM access to all of these files will in turn give users access to the interdepartmental information within in through a much more streamlined process. Files in the Long-term memory, as well as their corresponding embeddings, should be stored on databases and vector databases respectively that are accessible by the LLM. These files can be easily queried afterward when creating the prompt context, as well as when fine-tuning the model.

Sensory data is mostly composed of live or very recent data, be it either numerical or text data streams. This live or recent data will mostly be generated in real-time from machinery and other equipment, for the purpose of keeping the LLM up to date on the exact current state of the environment. Sensory data is also stored for later analysis, and to help in generating Long-term memory data.

For both Long-term memory files and Sensory data, most data files come with both numeric and text data, which are handled the same way by the LLM provided that they have the appropriate contextual information, in which case it should be able to classify and use the data in a meaningful way. However, in some cases, data may need to be reformatted or labeled in order to provide context that is otherwise missing within their respective file.

Lastly, the Short-term memory is comprised of the information that will be generated from the Conversation User Interface. More specifically, for one given conversation, the Short-term memory contains all the previous prompts and responses of that conversation, as well as the current prompt from the user. By having the entire chat history of a conversation, the LLM can now understand further inquiries without having to repeat the entire context of the conversation every prompt, allowing the model to give more accurate answers in return.

Table 2. Examples of files that can be used by a LLM application, showcasing the fact that it can handle data of broadly different types and source departments. These are just a few examples, and LLMs are not limited to just the ones shown here.

Data Source	Data Classification	Department
Pay Stubs & Salary Documents	Files	Accounting
Budget Information	Files	Accounting
Accounts Payable / Receivable	Files	Accounting
Invoices	Files	Accounting
Product Warranty Data	Files	Engineering
Equipment Warranty Data	Files	Engineering
User and Maintenance Manuals	Files	Engineering
Company Info	Files	Executive
Equipment Maintenance Logs	Files	Facility Management
Facility Maintenance Logs	Files	Facility Management
Staff Information	Files	Human Resources
Inventory Logs	Files	Manufacturing Floor
Process Flow Data	Files	Manufacturing Floor
Live Sensor Data	Data Streams	Manufacturing Floor
Sensor Logs	Files	Manufacturing Floor
Standard Operating Procedures	Files	Manufacturing Floor
Quality Data	Files	Manufacturing Floor/ Engineering
Process Control Data	Files	Manufacturing Floor/ Engineering
Live Machine Control Code	Data Streams	Manufacturing Floor
Material Pricing Data	Files	Purchasing
Material Order Contracts	Files	Purchasing
Raw Material Order Information	Files	Purchasing
Sales Contracts	Files	Sales
Order Logs	Files	Sales
Sales Data	Files	Sales
Shipping & Logistics Info	Files	Shipping

4.4. Context Layer

For any prompt given by a user, the Prompt Context is the combination of all the information that the LLM must be given for it to generate an appropriate response to a user's request. First, some static context should be defined by the user so that the LLM always introduces this information into any

conversation's context. We will elaborate on this momentarily, but the information in the static context is relevant to the task orchestrator to make it aware of certain capabilities. Additionally, the expected behavior and functionality of the LLM should be explicitly stated to the model itself through the user, guaranteeing it will be available in the prompt context. This should include a general idea of what information to expect, what format or types of responses the model should aim to give, and the tone of the responses. A description of the behavior the model should emulate allows it to work more accurately than it would without this context. As mentioned earlier, it is important to explicitly inform the Task Orchestrator of the available external agents it has access to, by describing the functionality and format of interaction in the static context which will be stored in prompt context. This makes the Task Orchestrator aware of external agents, their purpose, and how it should generate the correct function or API calls to these agents in order to gather more information from them.

Aside from the static context, the Short-term memory, Sensory Data, and Long-term memory are also included as part of a prompt's context. In the case of Short-term memory and Sensory data, they are always treated as context in the prompt and should be automatically included in their entirety every time a question is asked. For Long-term memory, since including the equivalent of entire databases in every prompt would be unfeasible, we use the contents of the Short-term memory to obtain the most relevant information from the Long-term memory so that can be added to the prompt context. This is done by creating embeddings in the Short-term memory and using them to obtain the files and data in the Long-term memory which are most closely related to the actual prompt. From all of this information, static and dynamic, we create the complete prompt context, which is then passed onto the Task Orchestrator to attempt to answer a request.

There are some cases when we would need to alter the behavior of our model to an even deeper level, which is where we then move into fine-tuning the language model. Model Fine Tuning is the process of adjusting the pre-trained model's internal parameters on new or specific data to improve its performance on a particular task. At a high level, this process involves preparing and uploading training data, executing training in accordance with selected hyperparameters, evaluating the performance of the resultant model, and iteratively refining these steps until reaching an optimally fine-tuned model. Notably, fine-tuning LLMs encounters formidable challenges due to the insufficient VRAM and computational capabilities of standard desktop computers, yet alternatives like cloud services and other pay-for-computing options provide access to this capability. Despite the potential merits of fine-tuning for specialized applications, an initial focus on refining prompt engineering is advisable due to its less resource-intensive nature, which requires significant time and effort. Fine-tuning emerges as particularly advantageous when a pre-trained model persistently underperforms, be it through failing to yield the desired answer, generating outputs in incorrect formats, or not fully adhering to given prompts, even after considerable refinement of the static information of the Prompt Context. Crucially, documenting these instances of failure for subsequent training or evaluation of a fine-tuned model is vital, capturing the essence of where the model's responses deviate from expectations. This documentation should encompass the prompt, the model's inadequate response, and the ideally envisioned response, formatted according to the specific requirements of the model being fine-tuned. Our framework accentuates the pivotal role of fine-tuning in enhancing model performance, yet, reiterates the primacy of prompt engineering as the foundational step for augmenting the efficacy of pre-trained models, advocating for a strategic blend of both methodologies to achieve better results.

4.5. LLM Layer

For this application framework, we define two main specialized Large Language Models: The Task Orchestrator and the Conversational Agent. All the information compiled and combined with the initial question is processed by the Task Orchestrator.This LLM is specialized to use the knowledge provided in the prompt context as well as its own trained knowledge, to generate and iterate over a list of tasks, in which it will perform back-and-forth communication with the external agents through function or API calls as it deems needed and possible, or pass the current accumulated information to the Conversational Agent. The Conversational Agent is specialized to take the resulting data as a new prompt and to generate a human-readable response, which can either answer the user's query or request further information as necessary. This final response from the Conversational Agent is fed back into the active chat context and used as data for any additional questions that may be asked in the conversation.

4.6. External Agents Layer

Tasks that are not immediately able to be answered, such as those that require further processing and analysis, are delegated to external agents. These agents can range from other AI models such as computer vision models, classification models, mathematical models, statistical models, and/or other functional algorithms.

As mentioned earlier the Task Orchestrator agent is made aware of the existence and functionality of these external agents through the static Prompt Context. It should contain specifications defining what APIs are available to the Orchestrator, as well as the format required to use them. Therefore, the Orchestrator possesses the capabilities to create a list of tasks that are mindful of the external agent's added value and can generate the necessary API calls to interact with them either directly or through an intermediate agent, depending on the actual implementation of the application. When faced with a prompt, the Orchestrator will communicate back and forth, either one by one or in parallel (depending on implementation), with these external agents going through

its defined list of tasks, until it reaches a satisfactory answer or it deems it necessary to request more information to the user, a task that is then passed onto the conversational agent.

It is important to keep in mind that the LLM application will always trust the results given by external agents unless specifically told otherwise in the prompt context or taught through model fine-tuning. As such, one should not expect in most cases for the LLM to recognize wrong answers from external agents, this validation should be done individually on the agents themselves.

5. Case Study

5.1. Milling Machine Case Study

To showcase the potential of an LLM assistant in a manufacturing context, as well as an application of the proposed framework, a chat-bot assistant was developed. It was created using the Streamlit Python library for its custom user interface which is hosted on a local computer as a web app. Hosting the model locally also allowed the web app to use the local computer's storage as the Long-term Memory, as well as to utilize the program's memory to store the active chat content (Short-term Memory). Additionally, we utilized several other tools such as the LangChain python library for its text splitting and vectorstore functions, OpenAI Embeddings API for embedding the text chunks into the vectorstore vector database, and OpenAI's GPT 3.5 API as our conversational agent. The chat-bot does not possess access to any external agents or a task orchestrator model. Therefore, the model's knowledge is limited to the information given to it through the complete prompt context, which is provided through a static prompt context (brief description of the desired behavior of the LLM), user prompts, and files uploaded to the web app. Finally, the model was not fine-tuned beyond it's original state, as the prompt context provided was enough for the model to behave desirably and return appropriate responses.

To present an example of the functionality of this application, the chatbot was tasked with assistance surrounding the use of a HAAS Desktop Mill. For this example, the Operator's Manual [25] of the machine was uploaded into the application, and the virtual assistant, which we denominated FILLIS (*Factory Integrated Logic and Language Interface System*), was then prompted to provide information based on the contents of this file. The size of the manual was 18.5 megabytes (MB), a total of 187 pages, and took an average of 14.78 seconds to be embedded with an average query response time of 4.74 seconds, where FILLIS was prompted to "provide a step-by-step tutorial on how to start up the HAAS Desktop Mill" (as shown later in the chat contents). Additionally, to investigate how file size affected performance, we uploaded an additional 6 documents to FILLIS' Long-term memory, all operator's manuals for different equipment, making the total database size 83 MB. This resulted in an average embedding time of 61.53 seconds (when embedding all files) with an average query response time of 4.11 seconds. In both cases the responses contained the same information (correct step-by-step instructions were provided) with only small formatting changes on each response. Returning to the chatbot, FILLIS was asked 4 different questions regarding the HAAS Desktop Mill. Two of these questions were possible to answer using the contents of the manual provided, one question was intentionally not answerable with only the manual as a knowledge base, and the last question asked FILLIS to translate a previous answer. The following is an extract from the chat contents for this example.

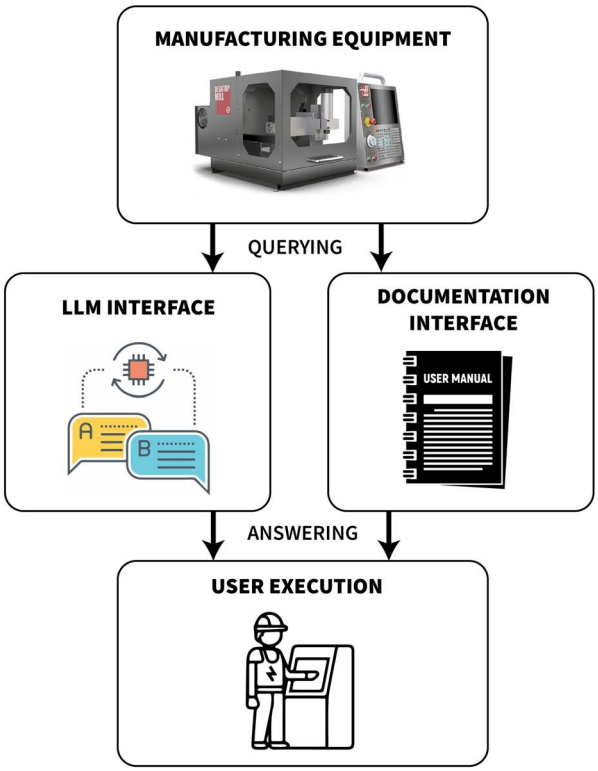


Figure 2. Flow diagram of the processes in the case study and the standard alternative for querying machine-specific information

Context: 2023 HAAS Mill Operator's Manual PDF [25]

User: "Give me a step by step tutorial on how to start up my HAAS Desktop Mill"

FILLIS: "Sure, here is a step-by-step guide on how to start up your HAAS Desktop Mill:

1. Turn on the Remote Display by following the instructions provided. Set a strong password for the Remote Display and press F4 to apply the settings.
2. Open the VNC Viewer application on your computer.
3. Enter the IP Address of the HAAS control in the VNC Server field. Select Connect.

4. At the login box, enter the password you set for the Remote Display on the HAAS control. Select OK.
5. The machine display will now show on your computer screen.
6. After the screen appears, perform a self-test and boot sequence. The startup screen will then be displayed.
7. Press CANCEL to dismiss the startup screen.
8. Turn the EMERGENCY STOP to the right to reset it.
9. Press RESET to clear any startup alarms. If you are unable to clear an alarm, contact your Haas Factory Outlet (HFO) for assistance.
10. If your machine is enclosed, close the doors.
11. Press POWER UP to start up the machine. Be cautious, as automatic motion begins immediately.
12. After the first POWER UP, the axes will move toward their home positions. The machine will then search for the home switch for each axis to establish the machine home position.
13. You can press CANCEL to dismiss the screen or press CYCLE START to run the current program.
14. To enter jog mode, press HANDLE JOG.
15. Press the desired axis (+X, -X, +Y, -Y, +Z, -Z, +A/C, -A/C, +B, or -B).
16. You can change the increment speeds while in jog mode. Each click of the jog handle will move the axis based on the current jog rate. You can also use an optional Remote Jog Handle (RJH) to jog the axes.
17. Press and hold the handle jog buttons or use the jog handle control to move the axis.

That's it! Your HAAS Desktop Mill should now be started up and ready for operation."

User: "What Gcode Command would I use to spin the spindle counterclockwise at 1500 RPM"

FILLIS: "To spin the spindle counterclockwise at 1500 RPM, you should use the Gcode command: S-1500 M04 The S-1500 sets the spindle speed to 1500 RPM in the counterclockwise direction, and the M04 command turns on the spindle in the specified direction."

User: "How would I remove the spindle from my Haas Desktop Mill"

FILLIS: "I'm sorry, but I don't have enough information to provide a specific answer to your question. It is recommended to consult the Haas Desktop Mill's user manual or contact Haas Factory Outlet (HFO) for assistance in troubleshooting and repairing any errors on the machine."

User: "Translate this to Spanish: Sure, here is a step-by-step guide on how to start up your HAAS Desktop Mill: . . . " (rest of the prompt is the same as the response from the first question)

FILLIS: "Aqu´ tienes una gu´a paso a paso sobre cómo iniciar tu HAAS Desktop Mill:

1. Enciende la Pantalla Remota siguiendo las instrucciones proporcionadas. Establece una contraseña segura para la Pantalla Remota y presiona F4 para aplicar los ajustes.
2. Abre la aplicación VNC Viewer en tu computadora.
3. Ingresa la Dirección IP del control HAAS en el campo VNC Server. Selecciona Conectar.
4. En el cuadro de inicio de sesión, ingresa la contraseña que estableciste para la Pantalla Remota en el control HAAS. Selecciona OK.
5. La pantalla de la máquina se mostrará en la pantalla de tu computadora.
6. Después de que aparezca la pantalla, realiza una autoprueba y secuencia de inicio. Luego se mostrará la pantalla de inicio.
7. Presiona CANCEL para descartar la pantalla de inicio.
8. Gira la PARADA DE EMERGENCIA hacia la derecha para reiniciarla.
9. Presiona RESET para eliminar cualquier alarma de inicio. Si no puedes eliminar una alarma, contacta a tu Haas Factory Outlet (HFO) para obtener ayuda.
10. Si tu máquina está cerrada, cierra las puertas.
11. Presiona POWER UP para encender la máquina. Ten cuidado, ya que el movimiento automático comienza de inmediato.
12. Después del primer POWER UP, los ejes se moverán hacia sus posiciones de inicio. Luego, la máquina buscará el interruptor de inicio para cada eje para establecer la posición de inicio de la máquina.
13. Puedes presionar CANCEL para descartar la pantalla o presionar CYCLE START para ejecutar el programa actual.
14. Para ingresar al modo de desplazamiento, presiona HANDLE JOG.
15. Presiona el eje deseado (+X, -X, +Y, -Y, +Z, -Z, +A/C, -A/C, +B o -B).
16. Puedes cambiar las velocidades de incremento mientras estás en el modo de desplazamiento. Cada clic del mango de desplazamiento moverá el eje según la velocidad de desplazamiento actual. También puedes usar un Mango de Desplazamiento Remoto (RJH) opcional para desplazar los ejes.
17. Presiona y mantén presionados los botones del mango de desplazamiento o usa el control del mango de desplazamiento para mover el eje.

¡Eso es todo! Tu HAAS Desktop Mill ahora deber´a estar encendido y listo para su funcionamiento."

The test prompts were meant to test the capabilities of the model, and it performed well and gave accurate information on the questions it was supposed to answer. For the first question where it is asked for startup procedures, FILLIS consolidated instructions from multiple different sections from pages 77, 95, 184 to create an easy-to-follow set of instructions for machine startup. The power of LLMs is shown in this case, as FILLIS concisely provided very important information in an easily readable manner that normally would have been spread out over more than 100 pages in the machine's manual docu-

mentation. On the question it was not supposed to answer, it told the user that it did not have sufficient information to answer the question and referred the user to the proper sources to find it. On the translation request, the model provided an almost perfect translation, without a version of the manual in the requested language. Even without using the most current version of OpenAI's GPT API, FILLIS is still able to complete simple to moderately complex tasks without the assistance of other agents. It provides accurate and concise answers, and even in the event of not lacking the knowledge to provide complete information, it is capable of suggesting actions that would lead to the requested data.

5.2. Discussion

As a proof-of-concept and a comparison of task completion time, accuracy, and quality of data given, FILLIS was compared to using the search function on a PDF version of the machine manual and to manually sifting through the machine documentation using the table of contents (See Figure 2). To begin the comparison we first started by asking FILLIS how to startup the Desktop Mill and waited to obtain a response before recording the metrics mentioned previously. We proceeded to the second scenario where we had to find the machine documentation on a standard computer, open it, and use the PDF search function to find the startup procedures. In the last scenario, we went through the machine documentation by hand using the table of contents to find the startup procedures, simulating a physical copy of the manual. Moving forward to the results of the comparison, obtaining an answer from FILLIS was the fastest overall, with the PDF search taking a similar amount of time, and searching through the document manually took over 3 times as long as the others. Even in the context of a relatively simple task where the file location of the desired document was known to be on the computer, FILLIS gave a more informative answer that included information from other sections of the document that were also related to startup including how to startup through the VNC Viewer as well. Using the PDF search function, only directed to the basic startup procedures, which were relevant but could have been missing context if the individual was also interested in connecting through the VNC Viewer. The power of the LLM is further shown through the fact that any clarifying questions could also be asked directly to the chat-bot if there was a misunderstanding with the instructions rather than having to look for the answers in the document.

5.3. Milling Machine Case Study Limitations

Some tasks are not fit for FILLIS's current capabilities and do require external agents, as FILLIS alone is only capable of calculating a written text answer. Tasks that involve mathematical operations, even simple ones, require external help to execute these calculations. As an example, when given the task to balance an assembly line, FILLIS was unable to correctly conduct basic addition and would respond with a different number for the sum of tact times almost every single time it was prompted to do so. When tasked with the addition of 24 numbers that summed to 406.969, with 7 trials the LLM came up with a sum of 442.273, 448.425, 418.514, 620.36, 398.078, 401.363, and 464.663. Not only did the model not get the sum correct a single time, but it also gave a different incorrect value every time. This limitation provides evidence for the need for external agents when creating a versatile data assistant tool. Additionally, the simple comparison between the methods we performed requires further investigation, as there are many different cases and situations in which FILLIS's capabilities can be fully tested and compared to traditional data-searching methods.

6. Conclusions and future work

Utilizing the vast amounts of data now accessible within modern manufacturing presents both an opportunity and a towering challenge. The escalating complexity and volume of data necessitate innovative solutions to efficiently manage and interpret this information. The utilization of Large Language Models (LLMs) emerges as a pivotal tool in this context, given their robust capabilities in processing and understanding extensive datasets effectively. Our proposed framework introduces an approach to directly tackle the identified research gaps by demonstrating how LLMs can aid manufacturers lacking specialized expertise in analyzing information integral to their manufacturing systems. Additionally, by providing a structured and practical framework for LLM applications in manufacturing settings, this study begins to narrow the gap between the theoretical potential and practical application of LLM technologies.

From our case study with FILLIS, we showcase the application's adeptness in delivering precise and context-aware responses. However, the case study also demonstrated limitations, such as the requirement for external tools to perform niche tasks (e.g., complex mathematical computations), in which our framework proactively outlines the necessity and method for incorporating supplementary agents to achieve a more versatile LLM assistant. Looking ahead, it is imperative to delve deeper into integrating external agents with LLM-driven orchestration, refining prompt engineering techniques for manufacturing-centric issues, and exploring the application of LLMs to more intricate and demanding manufacturing tasks. These avenues for future research not only extend the utility of the presented framework but also open new pathways for the comprehensive integration of LLM applications within the manufacturing domain.

Acknowledgements

We are grateful to the "UGA Innovation Factory" research group and to Luis Eduardo Izquierdo for collaboration and discussion. Additionally, we'd like to thank the University of Georgia for providing the facilities and capabilities to conduct this research.

References

[1] V. De Simone, V. Di Pasquale, and S. Miranda, "An overview on the use of ai/ml in manufacturing msmes: solved issues, limits, and challenges," *Procedia Computer Science*, vol. 217, pp. 1820–1829, 2023.

[2] M. Bernabei, S. Colabianchi, F. Costantino, *et al.*, "Natural language processing applications in manufacturing: a systematic literature review," ... *SUMMER SCHOOL FRANCESCO TURCO. PROCEEDINGS*, 2022.

[3] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems: Recent advances and new frontiers," *Acm Sigkdd Explorations Newsletter*, vol. 19, no. 2, pp. 25–35, 2017.

[4] S. Colabianchi, M. Bernabei, and F. Costantino, "Chatbot for training and assisting operators in inspecting containers in seaports," *Transportation Research Procedia*, vol. 64, pp. 6–13, 2022.

[5] M. C. May, J. Neidhöfer, T. Körner, L. Schäfer, and G. Lanza, "Applying natural language processing in manufacturing," *Procedia CIRP*, vol. 115, pp. 184–189, 2022.

[6] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 157–169, 2018.

[7] Y.-H. Kuo and A. Kusiak, "From data to big data in production research: the past and future trends," *International Journal of Production Research*, vol. 57, no. 15-16, pp. 4828–4853, 2019.

[8] X. Wang, A. Liu, and S. Kara, "Machine learning for engineering design toward smart customization: A systematic review," *Journal of Manufacturing Systems*, vol. 65, pp. 391–405, 2022.

[9] J. F. Arinez, Q. Chang, R. X. Gao, C. Xu, and J. Zhang, "Artificial intelligence in advanced manufacturing: Current status and future outlook," *Journal of Manufacturing Science and Engineering*, vol. 142, no. 11, p. 110804, 2020.

[10] J. Lee, J. Ni, J. Singh, B. Jiang, M. Azamfar, and J. Feng, "Intelligent maintenance systems and predictive manufacturing," *Journal of Manufacturing Science and Engineering*, vol. 142, no. 11, p. 110805, 2020.

[11] S. Colabianchi, A. Tedeschi, and F. Costantino, "Human-technology integration with industrial conversational agents: A conceptual architecture and a taxonomy for manufacturing," *Journal of Industrial Information Integration*, vol. 35, p. 100510, 2023.

[12] T.-Y. Chen, Y.-C. Chiu, N. Bi, and R. T.-H. Tsai, "Multi-modal chatbot in intelligent manufacturing," *IEEE Access*, vol. 9, pp. 82118–82129, 2021.

[13] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Applications*, vol. 2, p. 100006, 2020.

[14] L. Liu and V. G. Duffy, "Exploring the future development of artificial intelligence (ai) applications in chatbots: A bibliometric analysis," *International Journal of Social Robotics*, vol. 15, no. 5, pp. 703–716, 2023.

[15] K. Pandya and M. Holia, "Automating customer service using langchain: Building custom open-source gpt chatbot for organizations," *arXiv preprint arXiv:2310.05421*, 2023.

[16] R. Mahadevan, R. C. Raman, *et al.*, "Comparative study and framework for automated summariser evaluation: Langchain and hybrid algorithms," *arXiv preprint arXiv:2310.02759*, 2023.

[17] A. Pesaru, T. S. Gill, and A. R. Tangella, "Ai assistant for document management using lang chain and pinecone," *International Research Journal of Modernization in Engineering Technology and Science*, 2023.

[18] B. Sai, S. Thanigaivelu, N. Shivaani, S. Babu, and A. Ramaa, "Integration of chatbots in the procurement stage of a supply chain," in *2022 6th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 1–5, IEEE, 2022.

[19] A. Moeuf, S. Lamouri, R. Pellerin, S. Tamayo-Giraldo, E. Tobon-Valencia, and R. Eburdy, "Identification of critical success factors, risks and opportunities of industry 4.0 in smes," *International Journal of Production Research*, vol. 58, no. 5, pp. 1384–1400, 2020.

[20] S.-V. Buer, M. Semini, J. O. Strandhagen, and F. Sgarbossa, "The complementary effect of lean manufacturing and digitalisation on operational performance," *International Journal of Production Research*, vol. 59, no. 7, pp. 1976–1992, 2021.

[21] A. Moeuf, R. Pellerin, S. Lamouri, S. Tamayo-Giraldo, and R. Barbaray, "The industrial management of smes in the era of industry 4.0," *International journal of production research*, vol. 56, no. 3, pp. 1118–1136, 2018.

[22] W. H. Knol, J. Slomp, R. L. Schouteten, and K. Lauche, "Implementing lean practices in manufacturing smes: testing critical success factors' using necessary condition analysis," *International Journal of Production Research*, vol. 56, no. 11, pp. 3955–3973, 2018.

[23] S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "A critical review of smart manufacturing & industry 4.0 maturity models: Implications for small and medium-sized enterprises (smes)," *Journal of manufacturing systems*, vol. 49, pp. 194–214, 2018.

[24] M. Grohe, "word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data," in *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 1–16, 2020.

[25] HAAS Automation, *Mill Operator s Manual*. HAAS Automation, 2023.