e-ISSN: 2550-164X

Available online at
https://jmeche.uitm.edu.my/browse-journals/special-issues/

Journal of Mechanical Engineering SI 14 2025, 39 – 59.

Journal of Mechanical Engineering

Locally Hosted Conversational Process Mining for Production Data via Graph-based Retrieval-Augmented Generation (GraphRAG)

Shunfann Wu[1, 2], Cheng Yee Low[1*], Jingye Yee[3], Nicolaj C. Stache[2], Tobias Schmieg[2], Yupiter H. P. Manurung[4]

[1]*Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, Johor, Malaysia*
[2]*Faculty of Engineering, Heilbronn University of Applied Sciences, Heilbronn, Germany*
[3]*Fraunhofer Institute for Mechatronic Systems Design IEM, Paderborn, Germany*
[4]*Faculty of Mechanical Engineering, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia*

ARTICLE INFO

Article history:
Received 20 March 2025
Revised 28 September 2025
Accepted 03 October 2025
Online first
Published 15 November 2025

Keywords:
Process mining
Large language models (LLMs)
Graph-based retrieval-augmented generation (GraphRAG)
Intelligent manufacturing
Conversational chatbot

DOI:
10.24191/jmeche.v14i1.5757

ABSTRACT

A secure, on-premises conversational process mining chatbot that enables manufacturing companies to analyze production event logs with natural language queries is presented in this paper. The system adopts a graph-based retrieval-augmented generation (GraphRAG) approach to process mining: PM4Py discovers the process from logs, which is converted into concise activity-, path-, and variant-level facts and stores them with the process graph in a graph database. A hybrid retriever and a lightweight cross-encoder reranker select focused evidence for a compact large language model (LLM), enabling accurate answers about flows, bottlenecks, and variants. A key contribution is the fully local, open-source design covering the embedding model, graph database, reranker, and LLM, to ensure the privacy of sensitive and confidential data. The architecture is detailed using the Active Structure methodology, and the deployment is demonstrated with the Analytics Canvas in a representative use case. The result is a practical, private way for manufacturers to ask questions of their data and act on the insights.

INTRODUCTION

In the context of Industry 4.0, intelligent manufacturing plays a key role in monitoring and controlling the manufacturing processes. Process mining (PM) has emerged as a crucial bridge between data science and process science, enabling organizations to gain deep process insights from the increasing volume of data generated by modern manufacturing systems (Yurtay, 2022). The development of PM and large language models (LLMs) presents an opportunity to optimize manufacturing processes. By applying these advanced

[1*] Corresponding author. *E-mail address*: cylow@uthm.edu.my
https://doi.org/10.24191/jmeche.v14i1.5757

technologies, organizations can gain deeper insights into their operations, identify inefficiencies, and tackle bottlenecks by implementing effective solutions (Vidgof et al., 2023).

The potential synergy between PM and LLMs is explored in this paper through the development of a conversational agent designed to elevate manufacturing process monitoring to new levels. A further aim is the advancement of intelligent manufacturing within the Industry 4.0 framework, with a focus on providing Small and Medium-sized Enterprises (SMEs) with practical, cost-effective methods for integrating smart technologies into their production workflows.

While large industries such as Airbus, ASML, and BMW often leverage advanced, deeply integrated systems like enterprise resource planning (ERP), supply chain management (SCM), and manufacturing execution system (MES), which provide a solid foundation of vertical integration for process mining (Lechner, 2020; Rozinat et al., 2009; Valencia-Parra et al., 2021). These solutions are frequently not feasible for SMEs due to high implementation costs, lack of advanced IT resources, and the need for specialized IT expertise (Gering, 2020). Many SMEs are still in the early stages of their digital transformation, having achieved a minimal level of digitalization, such as recording production data in job travellers. Furthermore, their data collection methods are often fragmented, with information stored in various disconnected systems (Kgakatsi et al., 2024; Naeem & Garengo, 2022). This indicates that many SMEs are not yet ready for a complete, top-to-bottom digitalization of their entire enterprise. Instead, it is more practical and accessible for them to start with standalone data analytics solutions focused on specific, high-impact areas, such as their production processes (Stertz et al., 2021). This creates a significant gap, as these companies possess valuable data but struggle to extract actionable insights from it.

To address this challenge, a conversational process mining chatbot that uses GraphRAG, specifically designed for production processes, is proposed. The chatbot prioritizes data privacy by utilizing a fully local, open-source architecture. It is designed to work with production processes extracted from job traveler to provide valuable production insights while keeping data secure. By using open-source tools and hosting the system locally, the solution ensures that sensitive production data remains protected, addressing common privacy concerns with external LLM services (He et al., 2018).

The remainder of this paper is organized as follows. The following section provides an overview of related work in process mining, large language models, and GraphRAG. The next part details the system architecture, hardware configuration, dataset, chunking strategy, and model selection. This is followed by a comprehensive evaluation of the chatbot's performance. The subsequent discussion explores the implications and limitations of the study and outlines future work. The paper concludes with a summary of key findings.

RELATED WORKS

Chatbots for process mining and manufacturing

The application of natural language interfaces and chatbots to make process mining more accessible is an active area of research. Early approaches often relied on rule-based systems. For instance, Burattin (2019) proposed Process Mining Bot, a rule-based chatbot for performing PM tasks via Telegram, a messaging platform, while Yeo et al. (2022) proposed a rule-based system process mining query engine that relies on carefully designed template rules. While innovative, these rule-based systems are often limited in their flexibility and struggle to handle complex or unexpected user queries, as they cannot cover every possible scenario (Thorat & Jadhav, 2020). On the other hand, Fontenla-Seco et al. (2023), proposed a conversational agent for declarative PM called C-4PM which used Declare4Py frameworks to handle declarative PM tasks, the end response was handled by OpenAI LLM via API, this proof-of concept does not fully implement open-source components, leaving the system exposed to potential data breach or leak (Maheswari et al., 2023). Barbieri et al. (2025) proposed an architecture to build an LLM-based natural

https://doi.org/10.24191/jmeche.v14i1.5757

language interface for PM. In the broader manufacturing context, similar rule-based chatbots have been developed for tasks like human-to-MES interaction (Beric et al., 2020; Do & Jeong, 2022) and monitoring Overall Equipment Effectiveness (OEE) (Loh et al., 2023). However, these systems often face challenges in understanding user nuances and lack deep domain knowledge (Rooein et al., 2020).

Large language models for process analysis

The integration of LLMs with PM is a rapidly developing field of research. A key area of exploration is the use of textual abstraction techniques to make process models digestible for LLMs. Studies by Berti et al. (2024) and Berti & Qafari (2023) have demonstrated how process models can be textualized and fed into LLMs using open-source tools like PM4Py. Work from Jessen et al. (2023) has shown that prompt engineering techniques, such as few-shot learning, Chain-of-Thought (CoT) reasoning, can enhance the performance of LLMs in answering domain-specific questions without the need for fine-tuning the models. To support this emerging area, recent work has also focused on establishing evaluation methods for LLMs on PM tasks (Berti et al., 2024).

Despite this promise, integrating LLMs into process mining presents several key challenges. A primary limitation is the restricted context window of LLMs, which can hinder their effectiveness when analysing larger event logs (Busch et al., 2023). Secondly, pre-trained LLMs often lack deep domain-specific knowledge, leading to issues such as hallucination, where the model generates inaccurate or misleading outputs (Kampik et al., 2025). Finally, data privacy is a critical concern, especially when using closed-source LLMs where sensitive company data is sent to external servers for processing (Berti et al., 2024; Maheswari et al., 2023).

Graph retrieval-augmented generation (GraphRAG) for context-rich retrieval

Retrieval-Augmented Generation (RAG) is a technique designed to enhance the capabilities of LLMs by augmenting their prompts with information from an external knowledge source (Lewis et al., 2020). This is a proven method to mitigate the risk of hallucination and ground LLM responses in factual data (Tonmoy et al., 2024). However, the effectiveness of traditional RAG can be limited when dealing with highly structured or relational data (Zhang et al., 2025). The primary challenges were the poor query understanding when having a semantic similarity search (Ling et al., 2025). This is because traditional RAG typically relies on semantic similarity search over a vector index, which may not fully capture the structural context or the complex relationships between data entities (Purohit et al., 2024). A query may be semantically similar to a chunk but miss the crucial relational information that defines its role in the overall process.

To overcome these limitations, GraphRAG was introduced as an alternative architecture. GraphRAG extends the traditional RAG pipeline by leveraging a graph database as the external knowledge source instead of a standard vector database. The retrieval performance in GraphRAG can be highly enhanced by using hybrid retrieval, which combines both vector similarity search and full-text search to find relevant information (Bratanic & Hane, 2025; Tjokro & Ady Sanjaya, 2024). Furthermore, the use of a cross-encoder as a reranker, placed after the retrieval step to re-evaluate the retrieved information, has also been shown to improve the quality of the final response generation (Han et al., 2025).

Use case scenario using Analytics Canvas

To guide the design of the conversational PM chatbot with GraphRAG, the Analytics Canvas developed by Kühn et al. (2018) was adopted. This semi-formal, visual framework provides a clear, structured way to describe an analytics use case scenario using an extendable graphical notion. The Analytics Canvas breaks the structure into five layers: Analytics Use Case, Data Analysis, Data Pools, Data Description, and Data

https://doi.org/10.24191/jmeche.v14i1.5757

Source. This layered breakdown provides a clearer understanding of the system's functionality. The Analytics Canvas of a production process is shown in Fig 1.

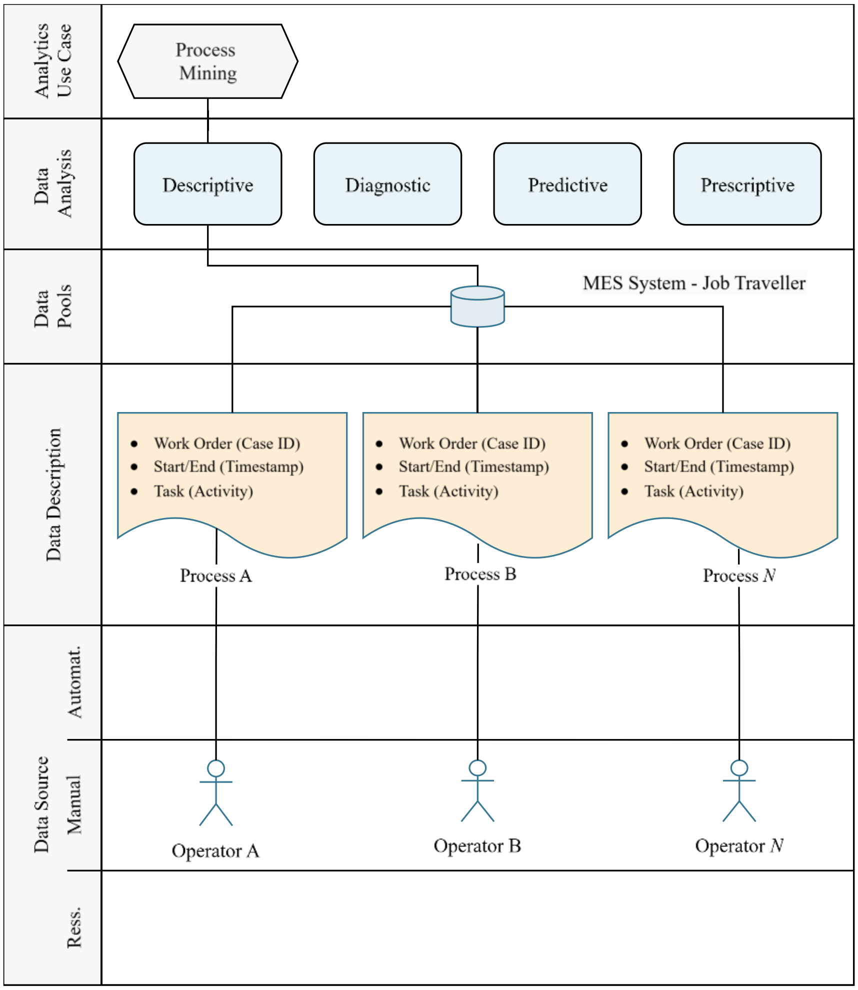

Fig. 1. Analytics Canvas for describing a process monitoring use case scenario with its infrastructural requirements.

First, the Analytics Use Case for the manufacturing processes is defined using the conversational PM chatbot. Next, operators from various stages in the manufacturing line act as the Data Source by scanning the job traveller each time they complete a task. Subsequently, the Data Description layer specifies the required data fields, ensuring clarity on the exact information needed for effective process analysis. In this scenario, three parameters were recorded using job traveller by operators through Processes A, B, and more. For each process, three essential parameters were recorded: work order IDs (case ID), start/end time (timestamp), and task (activity). Afterward, the Data Pools layer defines the storage and processing of the collected data. This data is gathered using job traveller, which is a key feature of MES software, and stored in a database. The raw data collected can then be used as event logs for executing PM and feeding into the conversational PM chatbot.

https://doi.org/10.24191/jmeche.v14i1.5757

RESULTS AND DISCUSSION

The system architecture is detailed in this section, employing an Active Structure, a systematic design methodology to develop a locally hosted, security-focused solution for manufacturing companies. This approach enables the extraction of actionable insights from manufacturing data without compromising data security. Additionally, the system outputs of the conversational PM chatbot with GraphRAG at different stages of the pipeline are presented in this section.

System architecture

The design and functionality of a conversational PM chatbot with GraphRAG are presented in the following section. The chatbot transforms production process data into valuable insights while ensuring data security through open-source solutions. The active structure of the conversational PM chatbot system designed in this study is shown in Fig 2. The relationships between the components are outlined in this architecture, detailing the flow of information from data sources to user interaction.

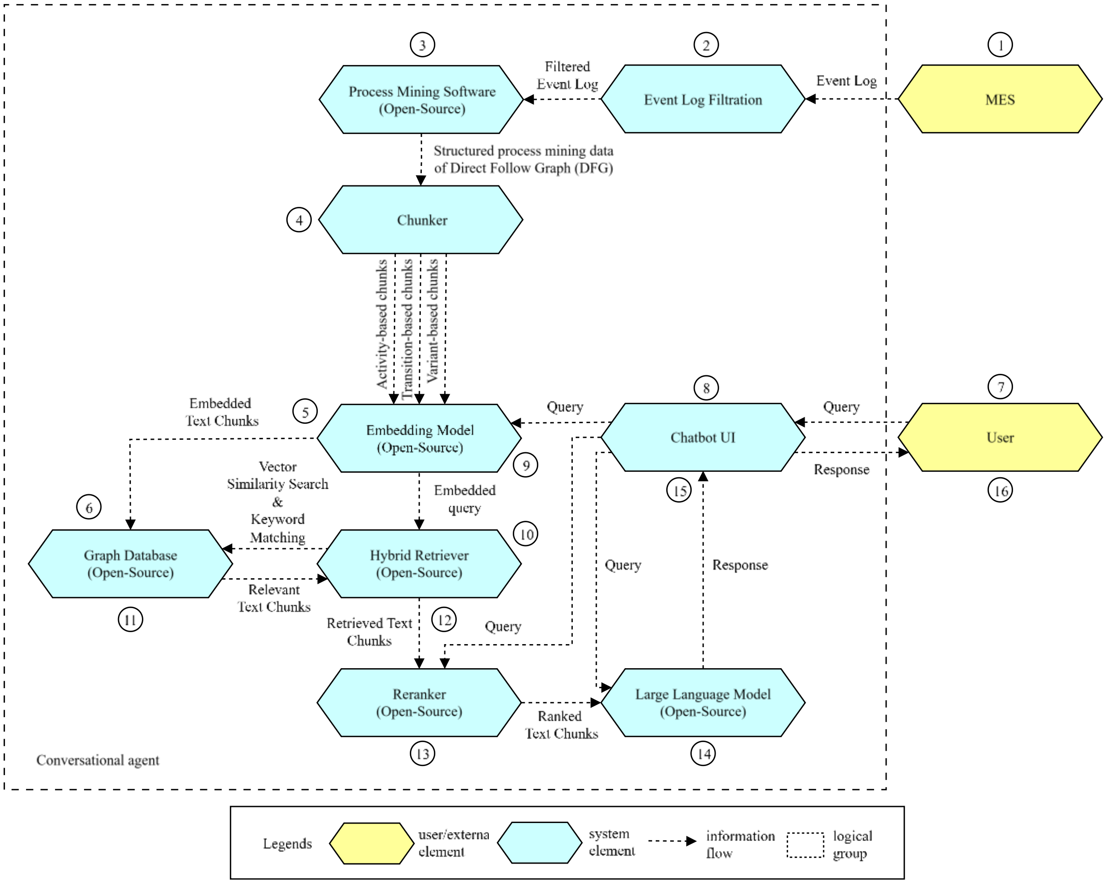

Fig. 2. The active structure of the conversational PM chatbot with GraphRAG.

(i) Manufacturing Execution System (MES): While the diagram conceptually shows MES as a data source, precisely the event log of the production floor, the current implementation of the system uses a publicly available raw production event log as the source (Levy, 2014).

https://doi.org/10.24191/jmeche.v14i1.5757

(ii) Event Log Filtration: The system processes the raw production event log, which involves delimiters handling, manual selection of production's part, and auto-detection of three core process mining columns: Case ID, Activity, and Timestamp. This ensures the event log is correctly structured for subsequent process mining analysis.

(iii) Process Mining Software: The prepared event logs are analysed using PM4Py, an open-source process mining library. PM4Py is used to discover the Directly-Follows Graph (DFG), identify start/end activities, and extract performance metrics and case variants. PM4Py provides structured data, and the system then generates natural language text chunks from this structured DFG data.

(iv) Chunker: The system employs an advanced chunking strategy. It generates three distinct types of natural language chunks from the PM4Py outputs: Activity-Based Chunks, Process-Based Chunks, and Variant-Based Chunks. These chunks are designed to capture different aspects of the process for efficient embedding and retrieval in the GraphRAG pipeline.

(v) Embedding Model: Each generated text chunk is converted into a high-dimensional vector embedding using a local, open-source embedding model. This process captures the semantic meaning of the text. The same embedding model is also used to embed user queries for similarity search. The local, open-source embedding model, Qwen3-Embedding-0.6B downloaded via Hugging Face was employed.

(vi) Graph Database: The vector representations (embeddings) of the chunks, along with their original text content, are stored in Neo4j, an open-source graph database. Neo4j is used not only to store the vector embeddings for semantic search (via its Vector Search Index) but also to store the interconnected process graph structure (DFG) itself and the text chunks that describe the activity and transitions.

(vii) User Query Input: Users interact with the system through a simple chatbot UI, submitting natural language queries to gain process insights.

(viii) Chatbot UI: It serves as the interaction point for users. It handles part name selection, receives queries, and presents the LLM-generated responses, providing a conversational interface to the backend system.

(ix) Embedded Query: User queries are converted into vector embeddings using the same local, open-source embedding model that was used for the chunks. This query embedding is then used for semantic similarity search in the retrieval phase.

(x) Hybrid Retriever: This component is a wrapper that wraps the hybrid Cypher retriever. It performs a search within Neo4j by combining vector similarity search and full-text keyword search. This allows it to retrieve not just relevant text, but also contextual information from the graph.

(xi) Semantic Similarity Search: Neo4j retrieves relevant embedded text chunks using its vector search index based on semantic similarity to the embedded query. This is part of a broader hybrid retrieval strategy that also incorporates full-text search and graph search.

(xii) Relevant Text Chunks Retrieval: The hybrid Cypher retriever retrieves relevant text chunks. These retrieved chunks are enriched with structured metadata directly from the process graph via dynamic Cypher queries. This context-rich information is then passed to the reranker.

(xiii) Reranker: The reranker is another type of language model that takes the relevant chunks retrieved by the retriever, along with the user query, and re-evaluates their relevance. This process is called reranking, it uses a more sophisticated language model, also known as cross

https://doi.org/10.24191/jmeche.v14i1.5757

cross-encoder to assign new scores, ensuring that only the most relevant information is passed to the LLM. The local, open-source reranker, ms-marco-MiniLM-L6-v2 downloaded via Hugging Face is used in this study.

(xiv) Large Language Model (LLM): The reranked context-rich information text chunks, along with the user query, are processed by a local, open-source LLM. The LLM combines this information along with a predefined system prompt to generate accurate and contextually relevant responses. The LLM, Qwen3-4B, 4bit quantized downloaded via Ollama is used in this study.

(xv) Response: The natural language response generated by the LLM is forwarded to the Chatbot UI.

(xvi) User: The Chatbot UI delivers the LLM's response back to the user, completing the conversational interaction loop.

Hardware configuration

The system developed in this study was deployed and evaluated on a local machine with the following specifications: an Intel Core i7-12700H processor, an Nvidia GeForce RTX 3060 Mobile GPU with 6 GB GDDR6 memory, 40 GB of RAM, and 512 GB of solid-state storage. These specifications are provided to ensure transparency and aid in the reproducibility of the experimental results. The project code is publicly available at https://github.com/amosshunfann97/PMChat.

Dataset description

A refined event log, derived from the public production event log (Levy, 2014), was processed for this study to focus on attributes relevant to manufacturing process mining: case_id, activity, timestamp, and part_desc. Each part_desc relates to one or more case_ids, with each case_id uniquely assigned to a part_desc. The dataset consists of 4533 events capturing production processes for various products on a manufacturing floor, recorded via a job traveler system and covering 42 unique part descriptions. For testing, only ‘Adjusting Nut’ (simpler) and ‘Plug’ (more complex) were analyzed for comparative purposes. Both parts were evaluated under identical system settings (retriever top-k = 15 chunks, reranker top-k = 5) to ensure a fair comparison of chatbot performance across varying process complexities. A sample of the datasets is presented in Table 1.

Table 1. A snippet of the refined event log dataset

case_id	Activity	Timestamp	part_desc
Case 103	Laser Marking - Machine 7	13-03-12 10:46	Adjusting Nut
Case 103	Flat Grinding - Machine 11	13-03-12 14:04	Adjusting Nut
Case 103	Final Inspection Q.C.	13-03-12 17:57	Adjusting Nut
Case 58	Laser Marking - Machine 7	19-02-12 16:24	Adjusting Nut
Case 58	Packing	20-02-12 1:00	Adjusting Nut
Case 121	Turning and Milling - Machine 4	14-03-12 23:32	Plug
Case 121	Turning and Milling Q.C.	15-03-12 9:38	Plug
Case 121	Laser Marking - Machine 7	16-03-12 11:39	Plug
Case 121	Deburring - Manual	18-03-12 8:17	Plug
Case 121	Flat Grinding - Machine 11	18-03-12 9:52	Plug

Source: https://github.com/amosshunfann97/PMChat/pmchatbot/Production_Event_Log.csv

https://doi.org/10.24191/jmeche.v14i1.5757

Chunking strategy

To effectively leverage LLMs for process mining insights, the raw process data is transformed into structured, natural language chunks. Following the process discovery and analysis performed by PM4Py, three distinct types of chunks are utilized, each designed to capture different aspects of the process model. These chunks serve as the knowledge base for the system. Illustrative examples of each chunk type are provided in Table 2Table 2, demonstrating how complex process information is transformed into a natural language format suitable for the GraphRAG pipeline.

Activity-based chunks

These chunks provide detailed descriptions of individual activities within the process. Each activity chunk includes information about its direct predecessors and successors, its execution count, and whether it is a start or end activity. Information regarding whether the activity has rework cases is also included. It also explicitly states if the activity has the highest or lowest execution count among all activities. For activities with self-loops, the chunk details the self-loop count and its rank among all self-looping activities.

Process-based chunks

These chunks describe an end-to-end process transition. Derived from frequent sequences of activities in the DFG, each process chunk outlines the sequence of activities in a particular path, its overall frequency, and performance metrics such as average, minimum, and maximum duration. These chunks also explicitly identify if a path is the fastest or slowest, has the highest or lowest frequency, and label the slowest paths as bottlenecks. The chunks also identify a self-loop pattern among the transitions.

Variant-based chunks

These chunks capture the diversity of process executions by describing unique case variants. Each variant chunk details the specific sequence of activities that constitutes a unique path taken by a process instance, along with its frequency of cases followed this exact path. It also includes the number of activities in each variant, its average, minimum, and maximum duration, and identifies any unique activities specific to that variant. The case_ids associated with each variant are also provided.

Table 2. Example of generated chunks

Chunk type	Description
Activity-Based	Turning & Milling - Machine 8 is an activity in this workflow. This activity has the highest/most frequency or execution counts among all activities. This activity was executed 10 times. Total number of activities: 7. Turning & Milling - Machine 8 is a starting activity that begins the workflow (1 cases start here). Turning & Milling - Machine 8 is preceded by: Turning & Milling - Machine 8 (8 times), Turning & Milling Q.C. (1 times). Turning & Milling - Machine 8 is followed by: Turning & Milling - Machine 8 (8 times), Turning & Milling Q.C. (2 times). This activity appears in 1 case. Case IDs linked to this activity: Case 103. This activity has rework cases. Turning & Milling - Machine 8 has a self-loop, directly followed by itself 8 times (rank 1/4 activities with rework within itself).
Process-Based	Process 'Round Grinding - Manual → Final Inspection Q.C.' is a transition that occurs 1 time. This transition takes on average 7 days 8 hrs 59 mins to complete. (min: 7 days 8 hrs 59 mins, max: 7 days 8 hrs 59 mins). This is the slowest/longest/highest transition in terms of execution time. This transition is the bottleneck due to its long average execution time. This transition has the lowest frequency or execution count out of 13 transitions.
Variant-Based	Process variant 'Round Grinding - Manual → Round Grinding - Manual → Round Grinding - Manual → Round Grinding - Manual → Final Inspection Q.C. → Laser Marking - Machine 7 → Packing' represents an execution pattern found in 1 case. Total number of variants: 2. This variant consists of 7 activities and takes on average 10 days 13 hrs 44 mins to complete. (min: 10 days 13 hrs 44 mins, max: 10 days 13 hrs 44 mins). This is the slowest variant. This is the shortest variant. This variant starts with Round Grinding - Manual, ends with Packing. This variant represents 50.0% among all variants. Case IDs following this variant: Case 58. All variants have the same number of execution count. This variant includes unique activities not found in other variants: Round Grinding - Manual.

https://doi.org/10.24191/jmeche.v14i1.5757

Model selection

The selection of appropriate large language models, embedding models, and rerankers is important for the overall performance, accuracy of the GraphRAG system. Open-source models were chosen to ensure a fully local and data-secured implementation.

For the LLM, which served as the conversational core, the system is configured to use a local instance of 4bits quantized Qwen3-4B served through Ollama. Despite its four-billion parameter footprint, Qwen3-4B demonstrates strong chain-of-thought reasoning capabilities in math and text. This capability allows it to surpass larger models, such as LLaMa-3.1-8B-Instruct, which has double the size and training parameters, in specific math and text reasoning tasks (Yang et al., 2025). This makes Qwen3-4B particularly suitable for process mining applications, as it can reason about concepts such as 'highest' and 'lowest' and perform numerical comparisons. This 'tiny-dense' reasoning model keeps GPU and CPU memory requirements within a single consumer-grade GPU card in an edge device, allowing for fully local deployment without external API calls. This model is responsible for generating natural language responses based on the retrieved context and user queries.

For semantic retrieval, Qwen3-Embedding-0.6B was chosen as the embedding model to embed both queries and graph chunks. Model exploration revealed that larger Qwen3-Embedding variants (4B and 8B) did not provide a significant retrieval performance boost, while leading to overloaded GPU memory and doubling inference latency. Moreover, Qwen3-Embedding-0.6B has demonstrated superior performance over other sub-1-billion models, such as BGE-M3 and even 7-billion alternatives in the MTEB embedding benchmark (Enevoldsen et al., 2025; Zhang et al., 2025). Its 600 million parameters effectively reduce runtime latency and memory usage, thereby offering the best balance between local-device workload and retrieval quality.

To refine the relevance of retrieved text chunks, the system utilizes the ms-marco-MiniLM-L6-v2 reranker. This language model, also known as a cross-encoder, has just 22 million parameters, adds negligible latency, yet is able to improve a more precise ranking of relevant chunks before the context reaches the LLM (Wang et al., 2020).

The careful selection of these models, balancing advanced capabilities with efficient resource utilization, ensures the GraphRAG system's effectiveness in providing accurate and contextually relevant process insights on a consumer-grade edge device, offering a robust, fully local, and data-secured GraphRAG pipeline for process mining applications.

Evaluation

A comprehensive evaluation of the conversational PM chatbot with GraphRAG is presented in this section, assessing its capability to accurately and effectively answer a range of process-related queries. The objective of this evaluation is to demonstrate the system’s practical utility in transforming complex event logs into actionable process insights through the GraphRAG architecture. The evaluation methodology aligns with the approach outlined in (Berti et al., 2024).

The questions posed to the chatbot were selected from a publicly recognized question bank, which contains 794 process mining questions. This question bank is publicly available at https://ic.unicamp.br/~luciana.barbieri/pmquestions.csv and has been utilized in previous expert evaluations of LLM’s capabilities in process mining (Berti et al., 2024; Barbieri et al., 2025). In this evaluation, a total of 15 queries were selected from the question bank, systematically categorized based on the type of process mining artifact they target: activity-based, process-based, and variant-based queries, reflecting the three chunking strategies implemented in the system. The evaluation criteria, the specific questions asked, and the chatbot's responses are detailed in the following subsections.

https://doi.org/10.24191/jmeche.v14i1.5757

Evaluation criteria

The performance of the conversational process mining chatbot with GraphRAG in generating responses was assessed using a three-tiered qualitative scale:

(i) Satisfactory (green): The chatbot is 100% accurate, complete, and presents the information from the “Ground truth” in a clear, concise, and natural language form. It effectively highlights key insights without errors.

(ii) Partially Satisfactory (yellow): The chatbot’s response carries the core information from the ‘Ground truth’ but has inaccuracies, incomplete results, or struggles with clarity and conciseness.

(iii) Unsatisfactory (red): The chatbot’s response is totally inaccurate, incomplete, and irrelevant to the query, failing to correctly interpret the data from the ‘Ground truth’. It may contain significant errors, contradictions, or hallucinations.

These evaluation criteria were applied to evaluate the chatbot’s generated results against the corresponding ‘Ground truth’ for each query presented in the following subsections.

Overall performance summary

A comprehensive overview of the conversational PM chatbot with GraphRAG's performance across all evaluated queries for both 'Adjusting Nut' and 'Plug' parts is provided in Table 3, presenting the satisfactory rating and generation time for each. Due to the length of the detailed results, including retrieval scores, rankings, and the full question and answer text, the complete evaluation data can be accessed via the GitHub repository provided in the Hardware Configuration section.

Activity-based queries (AQ)

The evaluation of the chatbot's performance on queries related to individual activities within the process is presented in this subsection.

For the questions “What are the most and least executed activities?”, "How many activities does the model have?", and "Which activity happens before and after activity X?", the chatbot was able to achieve a Satisfactory rating for both ‘Adjusting Nut’ and ‘Plug’ parts. This indicates that the system accurately and completely provided the requested information based on the queries. The ‘Plug’ generally exhibited 2-4 seconds longer generation time compared to ‘Adjusting Nut’, which can be attributed to the increased complexity a detail present in its corresponding process model and text chunks.

The high performance on these queries is largely due to the effective retrieval of relevant chunks. The activity-based chunks are designed to generate natural language descriptions that directly contain phrases that match the keywords and semantics of the questions. For instance, patterns such as “This activity has the highest frequency or execution count among all activities” are explicitly stated in the activity-based chunks, as shown in Table 2. This direct phrasing significantly aids the retriever in identifying and retrieving the correct chunk, which contributed to the chatbot’s accurate responses.

An example of the chatbot response for “What are the most and least executed activities?” is shown in Table 4Table 4. The chatbot's ability to accurately identify and present the correct activities and execution count for both process models is demonstrated in this example. The relevant information from the activity-based chunks is successfully extracted and formatted by the LLM into a clear and concise natural language response, including a listed format for enhanced readability, which earned a Satisfactory rating. The structured data provided as ground truth is directly reflected in the output, which confirms the effectiveness of the GraphRAG pipeline for this type of query.

https://doi.org/10.24191/jmeche.v14i1.5757

Table 3. Overall chatbot performance summary

Questions	Production parts			
	Adjusting Nut RERANKER_TOP_K=5 RETRIEVER_TOP_K=15		Plug RERANKER_TOP_K=5 RETRIEVER_TOP_K=15	
	Rating	Generation time (sec)	Rating	Generation time (sec)
Activity-based questions (AQ)				
What are the most and least executed activities?		6		9
How many activities does the model have?		3		4
What are the possible start and end activities in my log?		7		11
Which activity is the happened before and after activity X?		5		9
Which activities have instances of rework?		7		9
Process-based questions (PQ)				
What transitions take the highest average time and its frequency?		9		11
What are the main bottlenecks in the process?		12		13
What cases contain self-loops?		7		8
Where am I wasting the most time in the process?		7		11
Which are the highest and lowest transitions on average time and how frequent are they?		15		16
Variant-based questions (VQ)				
How many variants are there in the process?		2		5
Which cases followed the shortest path?		6		3
What is the longest trace regarding the number of activities? How many cases follow this trace?		6		9
How many distinct activities are there in the shortest path?		10		15
I would like to see statistics comparing longest and shortest variants.		15		13

Table 4. Example chatbot response for "What are the most and least executed activities?"

Question	Part	Ground truth	LLM generated result	Generation time
What are the most and least executed activities?	Adjusting Nut	- Turning & Milling - Machine 8: 10 executions - Flat Grinding -Machine 11: 1 execution	Most executed activity: Turning & Milling - Machine 8 (10 times) Least executed activity: Flat Grinding - Machine 11 (1 time)	6 secs
	Plug	- Turning & Milling - Machine 5: 18 executions - Grinding Rework- Machine 12: 1 execution - Round Grinding - Machine 23: 1 execution	Most executed activity: Turning & Milling - Machine 5 (18 executions). Least executed activities: Grinding Rework - Machine 12 (1 execution), Round Grinding - Machine 23 (1 execution)	9 secs

For questions such as "What are the possible start and end activities in my log?" and "Which activities have instances of rework?", the 'Adjusting Nut' part consistently received a Satisfactory rating. However, the 'Plug' part for these same questions received a Partially Satisfactory rating. While the chatbot's responses for 'Plug' were correct, they were incomplete. This limitation is attributed to the configured

https://doi.org/10.24191/jmeche.v14i1.5757

RERANKER_TOP_K setting of 5. Since information about each start, end, or rework activity is contained within its respective activity-based chunk, a query seeking a comprehensive list of such activities requires the retrieval of multiple relevant chunks. In cases where the total number of relevant activity chunks exceeded this top_k value, the reranker could only pass a subset of the complete information to the LLM, leading to a correct but shortened response. This highlights the impact of the top_k parameter on the completeness of responses when the number of relevant entities surpasses the RERANKER_TOP_K limit. The chatbot response for "what are the possible start and end activities in my log?" is presented in Table 5.

Table 5. Example chatbot response for "What are the possible start and end activities in my log?"

Question	Part	Ground truth	LLM generated result	Generation time
What are the possible start and end activities in my log?	Adjusting Nut	=== Start Activities === Turning & Milling - Machine 8: 1 case start here Round Grinding - Manual: 1 case start here === End Activities === Packing: 2 cases end here	Start activities: - Turning & Milling - Machine 8 (1 case starts here) - Round Grinding - Manual (1 case starts here) - End activities: - Packing (2 cases end here)	7 secs
	Plug	=== Start Activities === - Turning & Milling - Machine 5: 4 cases start here - Turning - Machine 5: 1 case start here - Turning & Milling - Machine 4: 1 case start here === End Activities === - Packing: 3 cases end here - Final Inspection Q.C.: 2 cases end here - Round Grinding - Machine 3: 1 case end here	Start activities: - Turning - Machine 5 (1 case starts here) - Turning & Milling - Machine 5 (4 cases start here) - Turning & Milling - Machine 4 (1 case starts here) - End activities: - Packing (3 cases end here) - Final Inspection Q.C. (2 cases end here)	11 secs

A Partially Satisfactory response from the 'Plug' part is illustrated in Table 6. The chatbot correctly identifies several start and end activities and their associated case counts. Specifically, for 'Plug' part, the LLM generated a correct response with accurate activity names and execution counts, but it listed only 5 activities instead of the total number of activities present in the ground truth. The impact of the RERANKER_TOP_K setting is directly demonstrated by this, which limits the number of chunks and, consequently, the amount of information passed to the LLM, resulting in an accurate but incomplete answer.

Process-based queries (PQ)

The chatbot's performance on queries focused on process flow, paths, and bottlenecks is evaluated in this subsection. For the questions "What transitions take the highest average time and their frequency?", "What are the main bottlenecks in the process?", "Where am I wasting the most time in the process?", and "Which are the highest and lowest transitions on average time, and how frequent are they?", the chatbot consistently achieved a Satisfactory rating for both 'Adjusting Nut' and 'Plug' parts. The system's strong capability in identifying key insights in the process-based chunks related to process performance and critical paths is demonstrated by this result.

Overall, questions in this category exhibited longer generation times compared to the AQ. This is attributed to the richer content of process chunks, which includes detailed time durations for each transition, and the increased reasoning from the LLM to synthesize responses for these types of questions. Consistent with previous observations, the 'Plug' part generally showed a 1-5 second longer generation time than 'Adjusting Nut', reflecting the greater complexity of its process model.

https://doi.org/10.24191/jmeche.v14i1.5757

Table 6. Example chatbot response for "Which activities have instances of rework?"

Question	Part	Ground truth	LLM generated result	Generation time
Which activities have instances of rework?	Adjusting Nut	=== Rework Cases Per Activity === Turning & Milling - Machine 8: 1 case with rework Turning & Milling Q.C.: 1 case with rework Final Inspection Q.C.: 1 case with rework Round Grinding - Manual: 1 case with rework	- Turning & Milling Q.C. - Final Inspection Q.C. - Round Grinding - Manual - Turning & Milling - Machine 8	7 secs
	Plug	=== Rework Cases Per Activity === Lapping - Machine 1: 6 cases with rework Turning & Milling - Machine 5: 5 cases with rework Round Grinding - Machine 3: 5 cases with rework Final Inspection Q.C.: 3 cases with rework Turning & Milling - Machine 4: 2 cases with rework Round Grinding - Q.C.: 2 cases with rework Round Grinding - Machine 12: 2 cases with rework Turning Q.C.: 1 case with rework Turning & Milling Q.C.: 1 case with rework Laser Marking - Machine 7: 1 case with rework Round Grinding - Machine 2: 1 case with rework Turning - Machine 5: 1 case with rework Round Grinding - Manual: 1 case with rework	- Grinding Rework - Machine 12 - Round Grinding - Machine 2 - Turning Q.C. - Round Grinding - Manual - Turning - Machine 5	9 secs

The chatbot's ability to accurately identify and articulate the primary time-wasting bottleneck in the process for both 'Adjusting Nut' and 'Plug' parts is demonstrated in Table 7. The relevant process-based chunks, which explicitly contain information about the slowest transitions and their characteristics, are successfully extracted by the LLM. For both parts, the LLM's response provided correct, clear details on the transition, its duration, frequency, and why it constitutes a bottleneck, which closely aligned with the ground truth. Crucially, this information is presented by the LLM in complete, natural language sentences, enhancing readability. A Satisfactory rating is earned by this performance, which highlights the effectiveness of the GraphRAG pipeline in providing actionable insights into process performance. Furthermore, the system's capability to semantically interpret user intent is demonstrated by the accurate identification of "wasting time" with the "slowest transition," which proves the advantage of having both a retriever and a reranker to understand the nuances of the query. Moreover, for the question "Which are the highest and lowest transitions on average time, and how frequent are they?", the chatbot also achieved a Satisfactory rating for both parts.

The system's capability to handle more challenging comparative queries is illustrated in Table 8. Information from both ends of the performance spectrum was successfully retrieved, and details for both the highest and lowest average were accurately identified and extracted. This demonstrates the effectiveness of the combination of retriever and reranker in providing the LLM with the necessary context to perform comparative reasoning.

https://doi.org/10.24191/jmeche.v14i1.5757

Table 7. Example chatbot response for “Where am I wasting the most time in the process?”

Question	Part	Ground truth	LLM generated result	Generation time
Where am I wasting the most time in the process?	Adjusting Nut	Round Grinding - Manual -> Final Inspection Q.C. (frequency = 1, performance = 637140.00 seconds = 7 days 8 hrs 59 mins)	- The slowest transition is **"Round Grinding - Manual → Final Inspection Q.C."** with an average execution time of **7 days 8 hrs 59 mins**, making it the primary time-waster. - This transition has the **lowest frequency** (1 occurrence) among 13 transitions, indicating it is a bottleneck.	7 secs
	Plug	Round Grinding - Q.C. -> Round Grinding - Machine 3 (frequency = 1, performance = 2337840.00 seconds = 27 days 1 hrs 24 mins	- **Waste in "Round Grinding - Q.C. → Round Grinding - Machine 3"**: - **Average execution time**: 27 days 1 hrs 24 mins (longest in the process). - **Frequency**: 1 (lowest among 43 transitions). - **Bottleneck**: Longest duration, causing significant time waste despite low frequency. - **Key Insight**: The most time wasted is in "Round Grinding - Q.C. → Round Grinding - Machine 3" due to its extreme execution time, even though it occurs infrequently.	11 secs

Table 8. Example chatbot response for “Which are the highest and lowest transitions on average time, and how frequent are they?”

Question	Part	Ground truth	LLM generated result	Generation time
Which are the highest and lowest transitions on average time and how frequent are they?	Adjusting Nut	Highest average time: Round Grinding - Manual -> Final Inspection Q.C. (frequency = 1, performance = 637140.00 seconds = 7 days 8 hrs 59 mins) Lowest average time: Final Inspection Q.C. -> Laser Marking - Machine 7 (frequency = 1, performance = 4380.00 seconds = 1 hrs 13 mins)	- **Highest average time**: Round Grinding - Manual → Final Inspection Q.C. (7 days 8 hrs 59 mins), frequency: 1. - **Lowest average time**: Final Inspection Q.C. → Laser Marking - Machine 7 (1 hr 13 mins), frequency: 1. - All listed transitions have a frequency of 1, which is the lowest among 13 transitions.	15 secs
	Plug	Highest average time: Round Grinding - Q.C. -> Round Grinding - Machine 3 (frequency = 1, performance = 2337840.00 seconds = 27 days 1 hrs 24 mins) Lowest average time: Lapping - Machine 1 -> Round Grinding - Manual (frequency = 1, performance = 960.00 seconds = 16 mins)	- **Highest average time**: - **Round Grinding - Q.C. → Round Grinding - Machine 3** (27 days 1 hr 24 mins) - **Frequency**: 1 - **Lowest average time**: - **Lapping - Machine 1 → Round Grinding - Manual** (16 mins) - **Frequency**: 1 All listed transitions have the lowest frequency (1) out of 43	16 secs

For the questions " Which transition contain self-loops?" the 'Adjusting Nut' part received a Satisfactory rating. However, the ‘Plug’ part for these same questions received a Partially Satisfactory rating, as shown in Table 9.

The only remaining "Partially Satisfactory" result in this category is for the question "Which transition contains self-loops?" for the 'Plug' part. This is attributed to the same RERANKER_TOP_K limitation observed in the AQ. Since information about self-loops is contained within individual process-based

https://doi.org/10.24191/jmeche.v14i1.5757

chunks, a comprehensive list requires retrieving more of the setting's relevant chunks. When the number of such chunks exceeds the top_k limit, the response becomes accurate but incomplete.

Table 9. Example chatbot response for "Which transition contains self-loops?"

Question	Part	Ground truth	LLM generated result	Generation time
Which transition contain self-loops?	Adjusting Nut	=== Self-loop Transitions (A -> A) === Total number of unique self-loop transitions: 4 Turning & Milling - Machine 8 -> Turning & Milling - Machine 8: 8 times Turning & Milling Q.C. -> Turning & Milling Q.C.: 1 time Final Inspection Q.C. -> Final Inspection Q.C.: 2 times Round Grinding - Manual -> Round Grinding - Manual: 3 times	- **Final Inspection Q.C. → Final Inspection Q.C.** (self-loop, 2 times) - **Round Grinding - Manual → Round Grinding - Manual** (self-loop, 3 times) - **Turning & Milling - Machine 8 → Turning & Milling - Machine 8** (self-loop, 8 times) - **Turning & Milling Q.C. → Turning & Milling Q.C.** (self-loop, 1 time)	7 secs
	Plug	=== Self-loop Transitions (A -> A) === Total number of unique self-loop transitions: 10 Turning & Milling - Machine 4 -> Turning & Milling - Machine 4: 7 times Round Grinding - Machine 3 -> Round Grinding - Machine 3: 6 times Turning & Milling - Machine 5 -> Turning & Milling - Machine 5: 9 times Round Grinding - Machine 2 -> Round Grinding - Machine 2: 3 times Round Grinding - Q.C. -> Round Grinding - Q.C.: 3 times Lapping - Machine 1 -> Lapping - Machine 1: 5 times Final Inspection Q.C. -> Final Inspection Q.C.: 5 times Turning - Machine 5 -> Turning - Machine 5: 1 time Round Grinding - Machine 12 -> Round Grinding - Machine 12: 4 times Round Grinding - Manual -> Round Grinding - Manual: 5 times	- Turning - Machine 5 → Turning - Machine 5 - Round Grinding - Machine 2 → Round Grinding - Machine 2 - Lapping - Machine 1 → Lapping - Machine 1 - Round Grinding - Machine 3 → Round Grinding - Machine 3 - Round Grinding - Manual → Round Grinding - Manual	8 secs

Variant-based queries (VQ)

The chatbot's performance on queries related to process variants, which represent the different execution paths within the process, is evaluated in this subsection. For all questions in this category, including "How many variants are there in the process?", "Which cases followed the shortest path?", "What is the longest trace regarding the number of activities? How many cases follow this trace?", "How many distinct activities are there in the shortest path?", and "I would like to see statistics comparing longest and shortest variants," the chatbot consistently achieved a Satisfactory rating for both the 'Adjusting Nut' and 'Plug' parts. This demonstrates the system's robust capability in understanding and articulating insights related to the diverse behaviours and patterns within the process.

The chatbot's strong capability in comparing and contrasting process variants based on specific criteria is presented in Table 10. For both 'Adjusting Nut' and 'Plug', the LLM successfully identifies the longest and shortest variants. Notably, the LLM interprets "longest" and "shortest" as referring to the total number of activities within the variant. However, it also extracts and presents relevant statistics such as activity

https://doi.org/10.24191/jmeche.v14i1.5757

count, duration, and associated case IDs from the variant-based chunks, allowing for a comprehensive comparison. The reasoning power of the LLM and the effectiveness of the GraphRAG pipeline in providing insightful variant analysis are highlighted by its ability to synthesize this comparative information, even for complex process models like 'Plug'. This performance earns a Satisfactory rating.

Table 10. Example chatbot response for "I would like to see statistics comparing longest and shortest variants."

Question	Part	Ground truth	LLM generated result	Generation time
I would like to see statistics comparing longest and shortest variants.	Adjusting Nut	Shortest variant: Variant 2 Number of activities in path: 7 Cases: 1 Case IDs: Case 58 Avg duration: 913440.00 seconds (10 days 13 hrs 44 mins) Longest variant: Variant 1 Number of activities in path: 19 Cases: 1 Case IDs: Case 103 Avg duration: 744120.00 seconds (8 days 14 hrs 42 mins)	- **Longest variant (activities): ** - **Variant: ** "Turning & Milling - Machine 8 → ... → Packing" (19 activities) - **Time: ** 8 days 14 hrs 42 mins (fastest, shortest time) - **Shortest variant (activities): ** - **Variant: ** "Round Grinding - Manual → ... → Packing" (7 activities) - **Time: ** 10 days 13 hrs 44 mins (slowest, longest time) - **Key difference: ** - Longest variant has more activities (19 vs. 7) but shorter duration. - Shortest variant has fewer activities but longer duration.	15 secs
	Plug	Shortest variant: Variant 2 Number of activities in path: 15 Cases: 1 Case IDs: Case 24 Avg duration: 3634560.00 seconds (1 months 12 days 1 hrs 36 mins) Longest variant: Variant 5 Number of activities in path: 26 Cases: 1 Case IDs: Case 122 Avg duration: 2030580.00 seconds (23 days 12 hrs 3 mins)	**Shortest variant**: - Activities: 15 - Duration: 1 month 12 days 1 hr 36 mins - Case ID: Case 24 - Unique activity: Round Grinding - Machine 23 - **Longest variant**: - Activities: 26 - Duration: 23 days 12 hrs 3 mins - Case ID: Case 122 - Unique activities: Round Grinding - Machine 2, Grinding Rework - Machine 12 - **All variants**: Same execution count. - **Both variants**: Start with "Turning & Milling - Machine 5" and end with "Packing".	13 secs

Overall, the robust capabilities of the conversational PM chatbot with GraphRAG in providing accurate and insightful responses across various process mining query categories are demonstrated by the evaluation. The system consistently achieved Satisfactory ratings for the majority of questions, showcasing its ability to semantically interpret user intent and translate complex process data into natural language. While some queries resulted in Partially Satisfactory ratings, these were primarily attributed to the configured top_k retrieval limits. A trade-off between completeness and computational efficiency is highlighted here, and it is suggested that finding an optimal top_k value or enriching the information density within each chunk could be areas for further improvement. The effectiveness of the GraphRAG pipeline in leveraging both the semantic power of LLMs and the structural context provided by the graph database for process analysis is validated by the successful performance across activity-based, process-based, and variant-based queries.

https://doi.org/10.24191/jmeche.v14i1.5757

DISCUSSION

An interpretation of the experimental results is provided in this section, highlighting the key findings, their implications for process mining, and the contributions of this study. The limitations of the current implementation are also addressed, and directions for future research are outlined. The robust capability of the conversational PM chatbot with GraphRAG in transforming complex event logs into actionable process insights was demonstrated by the evaluation. The system consistently provided accurate and contextually relevant responses across three query categories, including activity-based, process-based, and variant-based questions. The effectiveness of the implementation of the GraphRAG pipeline, which successfully integrates the semantic understanding of LLMs with the structural context provided by the Neo4j graph database, is emphasized by this performance. The ability of the system to semantically interpret user queries, such as identifying “wasting time” with “slowest transitions”, is a direct result of the combination of the retriever and reranker, which effectively identify and rank relevant information before passing to LLM. Furthermore, the system demonstrated a capacity for comparative reasoning, successfully identifying and contrasting entities from opposite ends of a performance spectrum, such as the highest and lowest average time transitions. Its advanced natural language understanding capabilities are further highlighted by this. Crucially, this entire system was developed employing an Active Structure, a systematic design methodology that ensures a fully local, secure-focused solution. In the manufacturing industry, protecting data security is a top priority (Mannhardt et al., 2019; Wong & Kim, 2017), especially with the increased risk of data leakage as companies embrace Industry 4.0 (Esposito et al., 2016). The choice of going for a fully local, open-source solution directly addresses concerns about data privacy leaks when using closed-source LLM service providers, where user queries are sent to external servers for processing (Berti et al., 2024; He et al., 2018). By integrating open-source components, the system is locally hosted, ensuring a secure, adaptable, and transparent approach to managing sensitive production data, making it particularly well-suited for environments where data privacy is the priority.

However, certain limitations were also highlighted by the evaluation. Queries requiring comprehensive lists of entities, such as queries asking for all start/end activities, rework, or self-loops, sometimes resulted in Partially Satisfactory responses when the process model became more complex. This was primarily due the the configured top_k both retriever and reranker limits, indicating a trade-off between completeness and computational efficiency. The performance of the retriever and reranker, while generally strong, can be constrained by these limits, affecting the completeness of the context provided to the LLM for reasoning and generating responses. Furthermore, while the system excels at extracting specific insights, requiring the LLM to regenerate a complete activity sequence for variants could potentially lead to hallucination and inaccuracies. The integration of GraphRAG has proven to be a practical strategy for handling complex, domain-specific data like event logs, and is particularly advantageous as traditional vector search methods are limited by their inability to utilize structural context and relationships between data (Purohit et al., 2024), a challenge effectively addressed by GraphRAG’s use of a graph database (Peng et al., 2024).

Future work will focus on addressing these limitations. Strategies to dynamically adjust top_k parameters or enrich the information density within each chunk will be explored to improve response completeness (Su et al., 2024). Further research will also explore alternative strategies for storing process mining data within the graph database. Investigating different graph schemas or data models could lead to more efficient and context-aware retrieval, further enhancing the performance of the GraphRAG pipeline. To enhance the conversational experience, integrating a memory mechanism that allows the chatbot to recall previous turns in a conversation will be explored. This would enable more natural, multi-turn dialogues and support more complex analytical workflows (Jessen et al., 2023). Given that the performance of retrieval is significantly impacted by the chunking strategy (X. Wang et al., 2024), further research into optimal chunking methods specifically for event logs is warranted to improve retrieval performance. Additionally, future studies will aim to explore its application to more advanced process mining tasks like conformance checking or predictive analytics, further enhancing its practical utility and generalizability.

https://doi.org/10.24191/jmeche.v14i1.5757

CONCLUSION

A conversational process mining chatbot with GraphRAG was successfully developed and evaluated in this study, demonstrating the practical utility of a novel GraphRAG architecture for extracting actionable insights from event logs. By leveraging open-source models and a locally hosted Neo4j graph database, the system provides a secure and self-contained solution for manufacturing companies, ensuring data privacy while delivering accurate and contextually rich process intelligence. The evaluation confirmed the chatbot's robust performance across various query types, validating the effectiveness of integrating semantic understanding with structural process knowledge. The significant potential of GraphRAG to democratize complex process analysis is highlighted by this work, which offers a powerful and accessible tool for industrial applications.

ACKNOWLEDGEMENTS/ FUNDING

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through GPPS (Q329).

CONFLICT OF INTEREST STATEMENT

The authors declare that this research was conducted without any personal, commercial, or financial conflicts of interest and confirm that there are no conflicting interests with the funding agencies.

AUTHORS' CONTRIBUTIONS

The authors confirm their contribution to the paper as follows. Conceptualization and methodology: Shunfann Wu, Cheng Yee Low, Jingye Yee; software development and testing, analysis and interpretation of results: Shunfann Wu; manuscript writing: Shunfann Wu, Cheng Yee Low; reviewing and editing the manuscript: Jingye Yee, Nicolaj C. Stache, Tobias Schmieg, Yupiter H. P. Manurung; supervision and funding acquisition: Cheng Yee Low. All authors contributed to every aspect of this work and have collectively reviewed and approved the final manuscript.

REFERENCE

Barbieri, L., Stroeh, K., Madeira, E. R. M., & Van der Aalst, W. M. P. (2025). An LLM-based Q&A natural language interface to process mining. In A. Delgado & T. Slaats (Eds.), Process Mining Workshops (pp. 5–17). Springer.

Beric, D., Havzi, S., Lolic, T., Simeunovic, N., & Stefanovic, D. (2020). Development of the MES software and integration with an existing ERP software in industrial enterprise. 19th International Symposium INFOTEH-JAHORINA (pp. 1–6). IEEE Publisher.

Berti, A., Kourani, H., Hafke, H., Li, C. Y., & Schuster, D. (2024). Evaluating large language models in process mining: capabilities, benchmarks, and evaluation strategies. In H. Van der Aa, D. Bork, R. Schmidt & A. Sturm (Eds.), Enterprise, Business-Process and Information Systems Modeling (pp. 13-21). Springer.

https://doi.org/10.24191/jmeche.v14i1.5757

Berti, A., & Qafari, M. S. (2023). Leveraging large language models (LLMs) for process mining (technical report) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2307.12701

Berti, A., Schuster, D., & Van der Aalst, W. M. P. (2024). Abstractions, scenarios, and prompt definitions for process mining with LLMs: a case study. In J. De Weerdt & L. Pufahl (Eds.), Business Process Management Workshops (pp. 427-439). Springer.

Bratanic, T., & Hane, O. (2025). Essential graphrag. Manning Publications.

Burattin, A. (2019). Integrated, ubiquitous and collaborative process mining with chat bots. Proceedings of the Dissertation Award, Doctoral Consortium, and Demonstration Track at BPM 2019 (pp. 144–148). CEUR-WS.org Publisher.

Busch, K., Rochlitzer, A., Sola, D., & Leopold, H. (2023). Just tell me: prompt engineering in business process management. In H. Van der Aa, D. Bork, H. A. Proper & R. Schmidt (Eds.), Enterprise, Business-Process and Information Systems Modeling (pp. 3-11). Springer.

Do, S., & Jeong, J. (2022). Design and implementation of RPA based ChatMES system architecture for smart manufacturing. WSEAS Transactions on Computer Research, 10, 88–92.

Enevoldsen, K., Chung, I., Kerboua, I., Kardos, M., Mathur, A., Stap, D., Gala, J., Siblini, W., Krzemiński, D., Winata, G. I., Sturua, S., Utpala, S., Ciancone, M., Schaeffer, M., Sequeira, G., Misra, D., Dhakal, S., Rystrøm, J., Solomatin, R., … Muennighoff, N. (2025). MMTEB: massive multilingual text embedding benchmark. International Conference on Learning Representations (ICLR) 2025 (pp. 1-57). OpenReview.net Publisher.

Esposito, C., Castiglione, A., Martini, B., & Choo, K. K. R. (2016). Cloud manufacturing: security, privacy, and forensic concerns. IEEE Cloud Computing, 3(4), 16–22.

Fontenla-Seco, Y., Winkler, S., Gianola, A., Montali, M., Lama, M., & Bugarín-Diz, A. (2023). The droid you're looking for: C-4PM, a conversational agent for declarative process mining. CEUR Workshop Proceedings, 3469, 112–116.

Gering, P. (2020). Digitalisation toolkit for SMEs. International Conference on Interoperability for Enterprise Systems and Applications (I-ESA) (pp. 58-65). Fraunhofer IPK Publisher.

Han, H., Wang, Y., Shomer, H., Guo, K., Ding, J., Lei, Y., Halappanavar, M., Rossi, R. A., Mukherjee, S., Tang, X., He, Q., Hua, Z., Long, B., Zhao, T., Shah, N., Javari, A., Xia, Y., & Tang, J. (2025). Retrieval-augmented generation with graphs [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2501.00309

He, F., Zhu, T., Ye, D., Liu, B., Zhou, W., & Yu, P. S. (2018). The emerged security and privacy of LLM agent: a survey with case studies. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 37(4), 111.

Jessen, U., Sroka, M., & Fahland, D. (2023). Chit-chat or deep talk: prompt engineering for process mining [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2307.09909

Kampik, T., Warmuth, C., Rebmann, A., Agam, R., Egger, L. N. P., Gerber, A., Hoffart, J., Kolk, J., Herzig, P., Decker, G., Van der Aa, H., Polyvyanyy, A., Rinderle-Ma, S., Weber, I., & Weidlich, M. (2025). Large process models: a vision for business process management in the age of generative AI. Künstliche Intelligenz, 39(2), 81–95.

Kgakatsi, M., Galeboe, O. P., Molelekwa, K. K., & Thango, B. A. (2024). The impact of big data on SME performance: a systematic review. Businesses, 4(4), 632–695.

https://doi.org/10.24191/jmeche.v14i1.5757

Kühn, A., Joppen, R., Reinhart, F., Röltgen, D., Von Enzberg, S., & Dumitrescu, R. (2018). Analytics Canvas – a framework for the design and specification of data analytics projects. Procedia CIRP, 70, 162–167.

Lechner, P. (2020). BMW: Process mining @ production: bringing innovation to production processes and beyond. In L. Reinkemeyer (Ed.), Process Mining in Action (pp. 65–73). Springer.

Levy, D. (2014). Production analysis with process mining technology. NooL - Integrating People & Solutions Publisher.

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. Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20) (pp. 9459–9474). Curran Associates Inc.

Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., Chowdhury, T., Li, Y., Cui, H., Zhang, X., Zhao, T., Panalkar, A., Mehta, D., Pasquali, S., Cheng, W., Wang, H., Liu, Y., Chen, Z., Chen, H., … Zhao, L. (2025). Domain specialization as the key to make large language models disruptive: a comprehensive survey. ACM Computing Surveys, 58(3), 1-39.

Loh, S. H., Jeuk, V. J., Sim, J. J., & Yeap, K. H. (2023). Implementation of artificial intelligence Chatbot in semiconductor manufacturing to optimize overall equipment effectiveness. 2023 IEEE 13th International Conference on Control System, Computing and Engineering (pp. 220–225). IEEE Publisher.

Maheswari, J. U., Vijayalakshmi, S., N, R. G., Alzubaidi, L. H., Anvar, K., & Elangovan, R. (2023). Data privacy and security in cloud computing environments. E3S Web of Conferences, 399, 04040.

Mannhardt, F., Petersen, S. A., & Oliveira, M. F. (2019). A trust and privacy framework for smart manufacturing environments. Journal of Ambient Intelligence and Smart Environments, 11(3), 201–219.

Naeem, H. M., & Garengo, P. (2022). The interplay between industry 4.0 maturity of manufacturing processes and performance measurement and management in SMEs. International Journal of Productivity and Performance Management, 71(4), 1034–1058.

Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., & Tang, S. (2024). Graph retrieval-augmented generation: a survey. Journal of the Association for Computing Machinery, 37(4), 111.

Purohit, S., Chin, G., Mackey, P. S., & Cottam, J. A. (2024). GraphAide: advanced graph-assisted query and reasoning system. IEEE International Conference on Big Data (pp. 3485-3493). IEEE Publisher.

Rooein, D., Bianchini, D., Leotta, F., Mecella, M., Paolini, P., & Pernici, B. (2020). Chatting about processes in digital factories: a model-based approach. Enterprise, Business-Process and Information Systems Modeling, 387, 70–84.

Rozinat, A., De Jong, I. S. M., Gunther, C. W., & Van Der Aalst, W. M. P. (2009). Process mining applied to the test process of wafer scanners in ASML. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 39(4), 474–479.

Stertz, F., Mangler, J., Scheibel, B., & Rinderle-Ma, S. (2021). Expectations vs. experiences – process mining in small and medium sized manufacturing companies. Business Process Management Forum, 427, 195–211.

Su, W., Tang, Y., Ai, Q., Wu, Z., & Liu, Y. (2024). DRAGIN: dynamic retrieval augmented generation

https://doi.org/10.24191/jmeche.v14i1.5757

based on the information needs of large language models [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2403.10081

Thorat, S. A., & Jadhav, V. (2020). A review on implementation issues of rule-based chatbot systems. Proceedings of the International Conference on Innovative Computing & Communications (pp. 1-6). Elsevier.

Tjokro, V. C., & Ady Sanjaya, S. (2024). Enhancing data traceability: a knowledge graph approach with retrieval-augmented generation. 2024 7th International Seminar on Research of Information Technology and Intelligent Systems (pp. 473–478). IEEE Publisher.

Tonmoy, S. M. T. I., Zaman, S. M. M., Jain, V., Rani, A., Rawte, V., Chadha, A., & Das, A. (2024). A comprehensive survey of hallucination mitigation techniques in large language models [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2401.01313

Valencia-Parra, Á., Ramos-Gutiérrez, B., Varela-Vaca, Á. J., Gómez-López, M. T., & García Bernal, A. (2021). Enabling process mining in airbus manufacturing: extracting event logs and discovering processes from complex data. Business Process Management Cases Vol. 2 (pp. 125–138). Springer.

Vidgof, M., Bachhofner, S., & Mendling, J. (2023). Large language models for business process management: opportunities and challenges [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2304.04309

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MiniLM: deep self-attention distillation for task-agnostic compression of pre-trained transformers. Proceedings of the 34th International Conference on Neural Information Processing System (pp. 5776-5788). Curran Associates Inc.

Wang, X., Wang, Z., Gao, X., Zhang, F., Wu, Y., Xu, Z., Shi, T., Wang, Z., Li, S., Qian, Q., Yin, R., Lv, C., Zheng, X., & Huang, X. (2024). Searching for Best Practices in Retrieval-Augmented Generation (No. arXiv:2407.01219). arXiv. http://arxiv.org/abs/2407.01219

Wong, K. S., & Kim, M. H. (2017). Privacy protection for data-driven smart manufacturing systems. International Journal of Web Services Research, 14(3), 17–32.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., … Qiu, Z. (2025). Qwen3 technical report [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2505.09388

Yeo, H., Khorasani, E., Sheinin, V., Manotas, I., An Vo, N. P., Popescu, O., & Zerfos, P. (2022). Natural language interface for process mining queries in healthcare. 2022 IEEE International Conference on Big Data (pp. 4443–4452). IEEE Publisher.

Yurtay, Y. (2022). Process mining in manufacturing: a literature review. Sakarya University Journal of Computer and Information Sciences, 5(3), 341–355.

Zhang, Q., Chen, S., Bei, Y., Yuan, Z., Zhou, H., Hong, Z., Chen, H., Xiao, Y., Zhou, C., Dong, J., Chang, Y., & Huang, X. (2025). A survey of graph retrieval-augmented generation for customized large language models [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2501.13958

Zhang, Y., Li, M., Long, D., Zhang, X., Lin, H., Yang, B., Xie, P., Yang, A., Liu, D., Lin, J., Huang, F., & Zhou, J. (2025). Qwen3 embedding: advancing text embedding and reranking through foundation models [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2506.05176

https://doi.org/10.24191/jmeche.v14i1.5757