UNIVERSITY OF TARTU
Institute of Computer Science
Innovation and Technology Management Curriculum

David Damola Obembe

Conversational interface for process mining tools using Large Language Models (LLMs)

Master's Thesis (20 ECTS)

Supervisor(s): Shefali Sharma
Marlon Dumas

Tartu 2024

Conversational interface for process mining tools using Large Language Models (LLMs)

Abstract:

Business process mining techniques allow business analysts to answer questions about the performance of business processes using data extracted from information systems. In existing process mining tools, business analysts often need to navigate across many diagrams and charts to find answers to their questions. Given the advancements in Large Language Models (LLM) for question answering, there is an opportunity to extend process mining tools to provide a conversational style to answer process mining questions. This study evaluates the ability of LLMs to answer process mining questions. The study evaluates two prompting approaches to answer process mining questions using LLMs. In the first approach (the direct approach), relevant business process performance metrics are included in the prompt. The LLM is asked to answer the question directly based on these metrics. In the second approach (the SQL approach), the LLM is told that there are tables available in a database to answer the questions. The LLM is asked to return an SQL query to answer each question. The study evaluates these two approaches using two LLMs, namely GPT-4 and Claude V3. The results show that these LLMs can answer process mining questions with precision and a recall ranging from 67% to 87%. Claude V3 slightly outperformed GPT-4 when using the direct approach, whereas GPT-4 performed better when using the SQL approach. The study also examines the effect of using XML tags to separate different sections of the prompt. This approach did not show any perceivable benefits.

Keywords:

business process mining, large language models, prompt engineering

CERCS:

P170 Computer science, numerical analysis, systems, control

Vestlusliides protsesside kaevandamise tööriistadele, mis kasutavad suuri keelemudeleid (LLM-id)

Lühikokkuvõte:

Äriprotsesside kaevandamise tehnikad võimaldavad ärianalüütikutel vastata küsimustele äriprotsesside toimivuse kohta, kasutades infosüsteemidest võetud andmeid. Olemasolevates protsessikaevandamise tööriistades peavad ärianalüütikud oma küsimustele vastuste leidmiseks sageli sirvima paljusid diagramme ja graafikuid. Arvestades edusamme suurte keelemudelite (LLM) alal, on võimalus protsessikaevandamise tööriistu laiendada, et pakkuda protsessikaevandamise küsimustele vastamiseks vestlusstiili. Selles magistritöös hinnatakse LLM-ide võimet vastata protsessikaevandamise küsimustele. Magistritöös hinnatakse LLM-ide abil protsessikaevandamise küsimustele vastamise kahte viipamise lähenemisviisi. Esimeses lähenemisviisis (otsene lähenemine) lisatakse viipesse asjakohased äriprotsesside toimivuse näitajad. LLM-il palutakse vastata küsimusele otse nende näitajate põhjal. Teises lähenemisviisis (struktuurpäringukeele ehk SQL-lähenemisviis) öeldakse LLM-ile, et andmebaasis on küsimustele vastamiseks saadaval tabelid. LLM-il palutakse igale küsimusele vastamiseks tagasi saata SQL-päring. Magistritöös hinnatakse neid kahte lähenemisviisi, kasutades kahte LLM-i, nimelt GPT-4 ja Claude V3. Tulemused näitavad, et need LLM-id suudavad vastata protsessikaevandamise küsimustele täpsusega ja meelespidamisega vahemikus 67% kuni 87%. Otsesel lähenemisel oli Claude V3 jõudluse poolest veidi GPT-4-st üle, samas kui SQL-i lähenemisviisis toimis paremini GPT-4. Magistritöös uuritakse ka XML-märgendite kasutamise mõju viipe erinevate osade eraldamiseks. See lähenemine ei näidanud märgatavat kasu.

Märksõnad:

äriprotsesside kaevandamine, suured keelemudelid, promptide inseneeria

CERCS:

P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

List of Figures

List of Tables

1 Introduction

In today's competitive landscape, businesses strive to continuously manage, improve, and optimize their business processes. According to Kaniški [1], a business process is a set of business activities sequentially arranged from start to finish in order to provide added value to their end users. The goal is to enhance operational efficiencies such that business activities are done faster, cheaper, and with fewer resources.

With this backdrop, large organizations now have dedicated roles, such as that of a process analyst, for identifying existing business processes, identifying areas of inefficiency, and proposing actions for improvements [2]. To do so, the process analyst uses a suite of techniques broadly called business process management (BPM) techniques [3]. These techniques range from process modeling to simulation, queueing analysis, waste analysis, and various process redesign techniques. Process modeling, for instance, enables organizations to visualize complex processes using models such as a Business Process Model and Notation (BPMN) diagram. This makes it easier to understand the flow of activities within the process. Among the plethora of BPM techniques, process mining has emerged in the past decade as a valuable tool to analyze business processes on the data stored in enterprise systems such as Customer Relationship Systems (CRMs) and Enterprise Resource Planning (ERPs) [4].

Process mining is a collection of state-of-the-art techniques for converting data extracted from ERP and CRM systems into actionable insight to create concrete business value. Extracting insights from process mining tools, however, requires that the process analyst has a solid understanding of various functionalities and involves a lot of manual work [5]. For instance, to use a modern process mining tool, the process analyst needs to inspect the process from different perspectives. The activity perspective is used to understand the workflow in the process, while the role perspective is used to understand handoffs between roles. Because a business process can involve hundreds or even thousands of activities, the process analyst needs to vary the levels of granularity during analyses. They determine the optimal granularity level to conduct their analysis [6]. However, it is also possible that when the level of granularity is reduced, some rework loops become lost, patterns are oversimplified, and findings are missed [7]. This is typically referred to as the spaghetti problem [8].

Apart from the spaghetti problem, process mining can also be time-consuming. Imagine that, in a claim-to-resolution process, a claims manager wants to know why certain claims are taking longer to process than expected and, as a result, are violating a service level agreement (SLA) of, say, five business days to respond to certain types of claims. While the end goal is to understand why they are not fulfilling a certain KPI, the first step is to understand the process. To achieve this, they need to filter cases that take too long, identify inefficiencies and bottlenecks in the process, and investigate possible reasons for the KPI violation. While existing process mining techniques have these capabilities, it takes a great deal of skill and effort [9].

Recent studies have shown that LLMs are great for question-answering tasks [10, 11, 12, 13]. Rather than the process analyst performing a number of steps before discovering the process, perhaps they could simply ask an LLM. This will enable process analysts to engage in conversations to quickly understand their processes rather than navigating through the complex maze of a process map.

In addition, current process mining tools can perform descriptive, diagnostic, and predictive analysis. However, they cannot recommend options for improvement. They reveal bottlenecks in the process but cannot directly answer the questions related to their options to improve the process. Recent studies have shown that LLMs contain a certain amount of information in their training data that allows them to potentially link a business problem with potential solutions [14].

For example, given that there is a problem of deadline violations and there are certain bottlenecks or rework in the process, LLMs may be capable of recommending options such as reducing the buffer times or the batching times. LLMs potentially allow us to connect insights from process mining, such as identifying bottlenecks, rework loops, and processing areas in the process to potentially improve the process. They could provide a way to connect the findings of process mining to improvements. Thus, it helps business process analysts address the weaknesses in their business processes with less effort.

1.1 Aim of the Study

The aim of the study is to understand the extent to which an out-of-the-box LLM model can perform process discovery and answer associated process optimization questions in business process mining.

1.2 Objectives of the Study

To achieve this aim, we need to configure and tune an LLM with appropriate inputs called prompts that will enable them to identify, interpret, or query the data generated by process mining techniques. To use LLMs in this context, we simply provide the LLM with the data generated by process mining and allow the user to ask questions about the process. We also ask questions about possible improvement opportunities. In this approach, we do not configure the LLM beforehand, a process called zero-shot learning. The first objective is:

Objective 1: To understand the extent to which LLMs can perform process analysis and process optimization using zero-shot learning.

We recognize the various LLMs are trained on different datasets. This will affect the performance of the LLMs on different tasks. In order to identify which LLMs perform the best on process mining tasks, the second objective of this study is:

Objective 2: To assess the relative performance of different LLMs, such as GPT-4[1] and Claude V3[2] , with respect to process mining tasks.

Studies have found that the power of LLMs lies in their prompts [15]. There have been various proposed techniques for altering the response of an LLM. The third objective of this study is thus:

Objective 3: To discover effective prompting strategies that improve the accuracy of LLMs for process analysis questions.

1.3 Research Method

This research was performed using the design science research (DSR) approach [16]. The problem this study aims to tackle was first scoped. Fundamentally, it was to understand how to use LLMs for process discovery and improvement, enabling a conversational approach towards process mining. To understand how LLMs can help achieve this, a literature review was conducted. The literature review studied the use of LLMs for process mining and prompt engineering techniques. Based on the literature reviewed, gaps were identified.

[1] https://openai.com/index/gpt-4-research/
[2] https://www.anthropic.com/news/claude-3-family

An initial prompting approach was designed as a solution to the problem stated. To evaluate this design, a set of process mining questions was curated, and the ground truth was defined. Given the information about the process performance statistics, the LLMs were asked to answer the process mining questions. The returned answer was evaluated against the ground truth to determine the measure of goodness. In addition, by applying guidelines from the field of prompt engineering, we engineered a set of prompts to improve the performance of the LLMs. We then re-evaluated the results, compared them to a baseline prompt, and evaluated the different alternative variations of those prompts.

The rest of the report is structured as follows. Chapter 2 explains preliminary concepts in business process management, process mining, LLMs, and prompt engineering. Chapter 3 provides a review of related work in this field. Chapter 4 describes the prompting method that could potentially be used for the experiments. Chapter 5 discusses the methodology adopted in performing the experiments, including the choice of LLMs used, the questions asked the dataset, and the evaluation criteria. Chapter 6 discusses the evaluation and findings of the experiments. Chapter 6 ties the findings from the experiments to the objective of this study and outlines directions for future work

2 Background

2.1 Business Process Management

Business process management (BPM) has become extremely important in understanding business processes. It helps organizations discover, model, analyze, measure, improve, and optimize work done to generate business value. To perform business process management, business process analysts need to first have a good understanding of the business process. This is the initial step of the BPM lifecycle — an approach for visualizing the phases in business process management. According to Dumas et al. [17], the BPM lifecycle is described in Figure 1

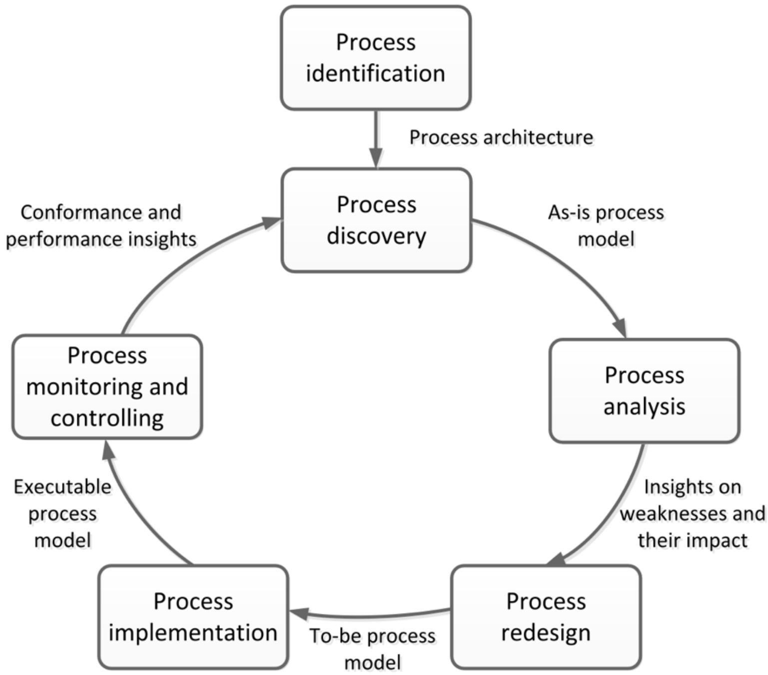

Figure 1: The BPM lifecycle [17]

2.1.1 BPM Lifecycle

Process identification is the initial phase of the BPM lifecycle. Here, the scope, objectives, and outcomes of the processes are identified by the process analyst. It establishes the processes that currently exist. After process identification, the process analyst needs to map out processes as they currently exist (called the 'as-is' state). This is called process discovery. Process discovery involves documenting the activities, transitions, and events in the process. The process can be visualized using a directly-follows graph (also called a process map) or a BPMN model (also called a process model).

The insights gathered during discovery are then analyzed in the process analysis phase. Here, the process analyst examines the 'as-is' process models, looking for bottlenecks, rework delays, and other inefficiencies. In the analysis phase, the process analyst attempts to perform root cause analysis to identify areas where the organization can make optimization efforts in terms of time, cost, and quality.

After process analysis is process redesign. The goal here is to transform the ‘as-is’ process model into a 'to-be' process model that addresses the bottlenecks and inefficiencies. After the redesign, the process analyst goes to the process implementation stage. This stage is about turning plans into action. It involves implementing the redesigned processes by updating systems, modifying roles, and training stakeholders to ensure they understand the new procedures.

After implementation, the organization's focus shifts to process monitoring and controlling. In this phase, the process analyst establishes systems to ensure the new processes are performing as intended. It's a continuous loop of measurement and adjustment where performance data feeds back into the system, allowing for real-time control.

Finally, from this continuous monitoring, we derive conformance and performance insights. This involves using the performance metric from process monitoring to further refine and optimize the process. Each phase of the BPM lifecycle is a step in a continuous cycle of business process improvement. The lifecycle circles back to the beginning, making the process iterative.

2.1.2 BPM Techniques

Traditionally, business process management involves extensive manual efforts [9, 17]. Process analysts would conduct interviews and workshops with employees across the various departments in the organization. Using responses from the interview, they gather insights about the business's data operations. The data is aggregated and gives them a bird-eye view of the processes within the organization. With this understanding, they then identify bottlenecks and develop possible solutions for optimization.

This approach is, however, time-consuming and could be a social desirability bias where employees do not reveal their actual activities during interviews [19]. To overcome these, modern techniques for business process management have evolved, utilizing digital tools, automation, and data. These techniques extract data from enterprise systems, such as CRM systems, in order to produce different types of visualizations that can help business process analysts identify weaknesses in their processes, understand reasons why certain KPIs are not being fulfilled, and discover opportunities for improving the process. One such modern technique is called process mining.

2.2 Process Mining

Process mining (sometimes called business process mining) is the art of extracting insight from event logs to understand how activities in the business are executed and identify areas for possible improvement. Business process mining ultimately helps businesses convert raw data from event logs to business value.

2.2.1 Event Logs

Event logs are digital records that capture the various activities that occur within an information system and the attributes of such activities. The attributes could include the timestamp of the activity, the role that performed the activity, the system that initiated the activity, and so on. Event logs can be derived from systems and applications within the system, such as ERP systems, CRM systems, supply chain management systems, web servers, workflow management systems, and so on.

Event logs are the backbone of process mining. The data from event logs are used to construct the sequence of events in the business process, forming a process map.

Table 1 shows an example of an event log in an ERP system.

Table 1: A sample event log from an ERP system.

Event ID	Event Time	Event Type	User	Description
1001	2023-02-19 10:00:00 AM	Order Created	Alex Li	Order #12345 created for 100 widgets
1002	2023-02-19 11:30:00 AM	Inventory Check	System	Inventory level for item #ABC123 is low (2 units).
1003	2023-02-19 14:00:00 PM	Purchase Order	Alex Li	Purchase order #PO12345 created for 50 units of item #ABC123
1004	2023-02-19 15:15:00 PM	Payment Received	Alex Li	Payment received for order #12345 (total: $1000).
1005	2023-02-19 16:00:00 PM	Item Shipped	System	Order #12345 shipped to customer

2.2.2 Process mining capabilities

Process mining enables users to perform a number of activities. The core activities include automated process discovery, performance mining, conformance checking, and variant analysis [20].

Automated process discovery is the first step in process mining, where algorithms are used to extract process models and maps from event logs [21]. This enables users to have an accurate understanding of the process using nodes and arcs. Building on that foundation, performance mining takes a closer look at the efficiency and timing of processes. It discovers the case duration, bottlenecks, and other performance statistics [22]. This is crucial for identifying inefficiencies enabling managers to make informed decisions on improvement opportunities.

Conformance checking further enhances process mining by comparing the actual behavior recorded in event logs against the expected models [20, 22]. This comparison helps identify deviations from business rules or compliance standards. Variant analysis focuses on the differences between instances of a process (also called a case) [24]. By understanding these variations, organizations can eliminate unwanted variants, leading to consistent quality outcomes.

Consider a scenario where a large organization seeks to optimize its global procurement process. The company uses automated process discovery to map out the procurement process as it currently operates across its various locations. Performance mining is then applied to identify bottlenecks, such as delayed approvals or lengthy supplier evaluations. Conformance checking reveals that some regional offices are not following the company's standardized procurement policies. Variant analysis is used to compare the performance of procurement processes across different regions, identifying best practices. With these process mining insights, the organization can implement clear and data-driven strategies that enhance its efficiency and competitive advantage.

While all the capabilities of process mining are vital, this thesis focuses primarily on automated process discovery and, secondarily, on performance analysis.

2.2.3 Automated process discovery

Automated process discovery is the analysis of event logs to reconstruct the sequence of events that occurred. This helps organizations better understand their operations and identify areas for improvement. Automated process discovery generates a process map, setting the stage for other process mining capabilities.

Process maps serve as a visual representation of the sequence of activities in a business process from start to finish [25]. They help business stakeholders understand the flow of activities, the frequency of each activity, and information across various stages of a process.

A process map consists of events, activities, and transitions.

- Events: These are actions that are done instantaneously. They act as triggers for initiating actions within the process flow.
- Activities: These are actual tasks or work performed in the process.
- Transitions: These connect activities and events, illustrating the flow from one to the next.

Figure 2 shows an example of a process map.

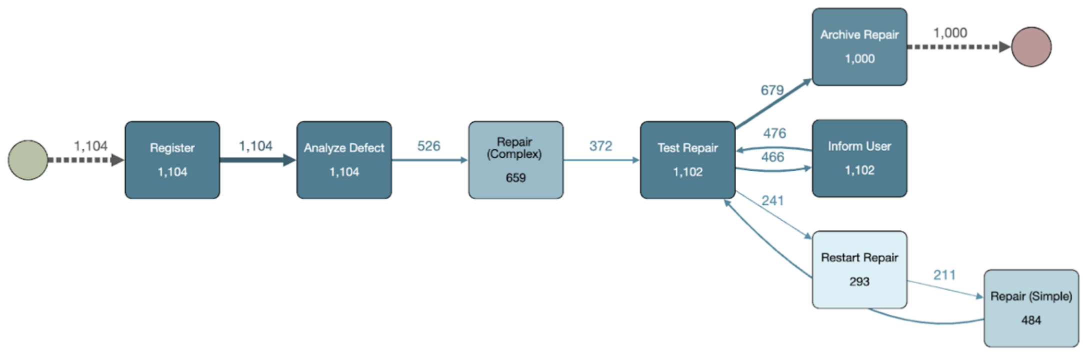

Figure 2: The process map of a repair process.

Process maps are useful for visualizing the process at a glance and identifying areas of improvement. For instance, a process map can help identify stages in the process where the flow is slower, causing delays and inefficiencies. Furthermore, a process map can help identify rework loops — an activity that is redone multiple times in a process instance. Identifying reworks helps reduce waste and improve the overall quality of the output.

In addition, process maps help uncover handoff delays — points in the process where delays occur due to the transfer of tasks from one resource to another. By visualizing the process flow, process maps make it easier to identify these handoff points, thereby enhancing operational excellence.

2.3 Large Language Models

Large Language Models (LLMs) are machine learning models trained on a massive amount of data. This allows them to perform a wide range of tasks, even beyond what was explicitly present in the training data [26].

LLMs are a significant advancement in natural language processing (NLP) that can understand texts to human standards. They are built on deep learning architectures with millions or even billions of parameters to understand and generate text that mimics human writing styles across various subjects [14].

2.3.1 Evolution of Language Models

In the beginning, models for textual understanding were simply rule-based [27, 28]. However, such models were limited. Machine learning started gaining popularity and was great for tabular data; however, it performed below par when faced with textual data [29].

Algorithms such as support vector machines (SVM), Word2Vec, and One-Hot encoding started to become the go-to models for machine learning applications with text — a field called natural language processing (NLP). In 2017, Google introduced a new network architecture called Transformers, which generalizes textual data well [30]. This led to the introduction of models like Generative Pre-trained Transformer (GPT) from OpenAI and Bidirectional Encoder Representations from Transformers (BERT) from Google, marking the beginning of a revolution in how machines understand and generate language [31]. These models are trained on large text, and because transformers are great at understanding context, these models had a good understanding of the nuances of language, including grammar and context [32].

2.3.2 Applications and Implications

LLMs have a wide range of applications. They have proven to perform tasks such as composing emails, generating articles, writing codes, etc. Their ability to understand and generate human-like text has made it gain application across industries, including education, healthcare, customer service, etc [31]. Many big tech companies now have their own LLMs. For instance, Google has Gemini[3], Microsoft has Copilot[4], OpenAI has ChatGPT[5], and Anthropic has Claude[6], to mention a few.

These companies now integrate LLMs with their products, enabling us to indirectly use artificial intelligence (AI) in our daily lives. For instance, the iPhone can now predict the next word to be typed when chatting. With Google Workspace, users can not summarize an email and craft a customized reply. Microsoft Copilot helps developers predict the completion of a class instantiation or a function based on its name.

While these applications look promising, there are concerns that it brings. Many people fear that LLMs may be out to take out our jobs [33]. There is also the fear of misinformation, as LLMs tend to blatantly hallucinate in some situations [34]. Rather than saying it does not know the answer, it could generate a wrong answer with unparalleled confidence. Beyond these, there is also the concern for bias in the model. Because the LLMs are trained on a vast amount of data, the data could contain some innate bias from humans. Initial models of GPT-3 were reported to be biased towards demography [35].

In addition, LLMs can be prone to adversarial attacks. Zou et al. [36] and Xu et al. [37] cited examples of these attacks in their papers. Xu et al. [37], discovered that LLMs can be tricked into producing undesirable outputs by asking it to repeat a single character or emoji multiple times.

2.4 Prompt Engineering

Prompt engineering is essential for unlocking the full potential of LLMs [38]. The prompts, which are the inputs given by the user, strongly impact the LLM response [39]. A number of prompt engineering techniques will be applied in this research. This section discusses

[3] https://gemini.google.com/
[4] https://copilot.microsoft.com/
[5] https://chat.openai.com/
[6] https://claude.ai/

various prompt engineering techniques, such as zero-shot learning, and discusses other best practices when using LLMs for process mining tasks.

2.4.1 Zero-shot vs Few-shot Learning

LLMs can be prompted using any of the three approaches: zero-shot, one-shot, and few-shot learning.

Zero-shot learning involves prompting the LLM to perform a task without providing any specific examples. The LLM relies on its pre-existing knowledge from its training across diverse datasets. In process mining, zero-shot learning is used when there's a lack of labeled data due to new or unpopular processes [40]. There are, however, situations where the LLM is provided with an example to learn from. When only one example is provided, this is called one-shot learning. The singular example acts as a reference from which the LLM is to take learnings when attempting its task. Few-shot learning expands on the one-shot approach by providing the AI with multiple examples. Few-shot learning has proven to improve the performance of GPT 3.5 and GPT 4 when performing process mining tasks [41].

2.4.2 Best Practices for LLMs in Process Mining

A number of literature have highlighted key learning points for performing process mining with LLMs. One of which is the importance of defining domain terms. For instance, if the definition of a bottleneck in the context of process mining is not provided in the prompt, the LLM may fall back to a generic interpretation of these words, which may not be accurate [41]. It is, therefore, important to provide the definition of domain-specific terms in the prompt to avoid ambiguity.

Furthermore, assigning a persona to an LLM, also called role-playing, improves contextual awareness, understanding of technique terms, and overall response [41]. For instance, the LLM can be called an expert software engineer, making it frame its responses in a manner that fits the title. Role-playing is a widely used technique when dealing with LLMs and has been shown to return more tailor-suited results to the specified persona [42].

Another best practice when working with LLMs is to explicitly ask the LLM to state its chain of thought. In other words, before the LLM performs a task, it first must state the steps it would take to accomplish this task. In Wei et al. [43], the authors showed a significant improvement in the reasoning ability of GPT-3 when asked to state its chain of thought. Jessen [41] also stated that adding the chain of thought allowed the LLM to have more precision and less hallucination.

In addition to the above, using XML tags to split the prompts into multiple segments is a great way of improving the LLM performance. The LLM pays attention to the start and end of XML tags and has a broader understanding of the prompt, particularly when it is large [44].

Another approach to improving the performance of LLMs is by implementing a so-called Retrieval-Augmented Generation (RAG). RAG is a technique where the LLMs use external information from the user to enhance their understanding of the context and provide more tailor-suited answers [45]. In process mining, the LLM can be linked to an agent that allows the LLMs to retrieve relevant context about the process from an external database. This can significantly enhance the contextual awareness of the LLM without consequently increasing the model's prompt size.

3 Literature Review

3.1 Related Work

In this section, an overview of related works is presented. The work discussed covers research in the intersection of process mining and large language models.

Large Language Models (LLMs) have ushered in new possibilities in various domains, including process mining [46]. Vidgof et al. [47] outlined some of the research opportunities for the use of LLM in BPM and proposed how LLMs can be applied in the various stages of the BPM lifecycle, including process identification, process discovery, and analysis. For instance, they asserted that LLMs can assess the health of a process, can extract patterns from logs, can act as a chatbot with domain expertise, and can identify problems within a process. While their work suggested directions for the potential application of LLMs in process mining, they did not implement these applications. This study will implement some of the opportunities put forward in their work, particularly in understanding the process and identifying problems within the process.

Grohs [46] explored the use of LLMs in the discovery phase of the BPM lifecycle. They used LLMs for three BPM tasks.

- For mining BPMN models.
- For mining declarative models.
- For assessing robotic process automation (RPA) potentials.

They used GPT-4 to classify process tasks into five Linear Temporal Logic (LTL) templates — precedence, response, succession, init, and end. The result showed GPT-4 providing consistent high-quality classifications. The authors also investigated whether LLMs can automate repetitive tasks. The LLM was prompted to classify tasks into manual, automated, or user tasks. The model excelled in identifying user tasks suitable for RPA, though with some variability in classification accuracy across different prompt executions. While this work relates to our overall study, it does not address the same application of LLMs.

LLMs have demonstrated proficiency in various coding tasks, such as code generation, code summarization, vulnerability mining, etc [48]. Following this theme, Jessen et al. [41] proposed an architecture for process mining analysis with LLMs. This involved adding the data schema to the prompt and requesting SQL queries to answer process mining questions. They loaded the event log as-is (containing fields such as case ID field, timestamp, activity, etc.) into a relational database. This data structure was passed into the prompt, alongside the definition of domain-specific terms and the question they require the LLM to answer [41]. They asked 795 process mining questions, such as:

"What are the root causes of the performance issues in the process? Please provide only process and data specific considerations, no general considerations."

GPT-3.5 and GGPT-4 were the LLMs used to evaluate questions in process discovery, conformance checking, advanced analysis, and so on. Their result showed that GPT 3.5 partially answered and fully answered 61 and 107 questions, respectively, while GPT 4.0 fully and partially answered 193 and 178 questions, respectively. Their work focused on GPT-3.5 and GPT-4. However, there have been new improvements to LLMs. OpenAI continuously rolls out update patches to GPT-4 [40], and Anthropic released Claude V3 with more training parameters in Q1 2024 [50]. This study will evaluate the performance of these new models.

Berti et al. [51] did a comprehensive review of how LLMs are being used for process mining. They detailed LLM capabilities for process mining, including a long context window, acceptance of visual prompts for interpreting process maps and models, coding capabilities for generating queries, and context understanding to ensure information accuracy.

Another approach to using LLMs in process mining is by abstracting event logs and process models in textual form. Because event logs can become large, it may be useful to concisely describe the process with its performance statistics in a small text. Berti et al. [10, 43] used the Pm4Py library to abstract DFGs and petri nets in a more concise text for LLMs. An example of an abstracted DFG using the Pm4Py library is shown below:

Activity A –>Activity B (frequency = 2 performance = 1350.0)

Activity C –>Activity D (frequency = 1 performance = 900.0)

Activity B –>Activity C (frequency = 1 performance = 900.0)

Activity A –>Activity C (frequency = 1 performance = 1800.0)

Where Activity A –>Activity B indicates that Activity B directly follows Activity A. Frequency is the number of times the transition occurred, and performance is the average duration of the transition in milliseconds.

Using publicly available event logs, they tested the LLM's capability to provide answers to process mining questions based on abstracted process mining data. While GPT-4 demonstrated proficiency in generating actionable insights from abstracted data, they noted challenges such as the unstableness of the LLM responses and privacy concerns with sensitive company data. Furthermore, the abstracted DFG used contained only two performance statistics: the frequency and performance (meaning the average duration). In this study, we will add a richer set of performance statistics discovered from Apromore's Process Discoverer, such as the relative case frequency, the average, median, minimum, maximum, and total rework, frequency, and duration.

3.2 State of the art

To review the state-of-the-art in the field of process management, we adopted the snowballing approach. The proceedings of the 21st International Conference on Business Process Management were reviewed, and related papers were identified.

Barbieri [53] proposed a proof-of-concept to build a conversational interface for process mining. The approach involved converting natural text to a logical form that can be mapped to a process mining interface. Given a question, they use native NLP approaches such as tokenization, part-of-speech (POS) tagging, lemmatization, etc, to pre-process the text. They then used rule-based semantic parsing to convert the text into clearer instructions for the processing mining interface. For instance, "What is the average execution time of the process?" becomes:

select case

project duration

aggregate duration

They also suggested the use of a text-to-speech interface such that users speak to the process mining interface rather than type the questions as text.

The conversation around LLMs extends beyond process mining into the broader domain of BPM, as explored by Raiaan et al. [32]. They studied the potential of LLMs in the six stages of the BPM lifecycle: identification, discovery, analysis, redesign, implementation, and monitoring. They, however, identified challenges with using LLMs for process mining, such as data complexity, real-time responsiveness, bias and hallucination, privacy concerns, and so on. Nevertheless, they emphasize the potential of LLMs to be integrated into commercial process mining tools.

Kecht et al. [54] discussed the importance of organizational chatbots understanding business processes. In their work, they investigated the problem of quantifying chatbots' ability to learn and adhere to organizational business processes. Using a Design Science Research Methodology (DSRM) approach that integrates conformance checking, they evaluated the chatbot's performance on adherence to business processes, using metrics such as lexical diversity and sentence BLEU scores. This approach can help business stakeholders understand how well their chatbots understand the business process and provide tailor-suited answers to users.

Kobeissi et al. [55] addressed the challenge of enabling domain analysts, who may lack database expertise, to query process execution data efficiently. In order to perform process-oriented data analysis, the user needed to understand database querying language. In a bid to democratize the field, the authors introduced a natural language interface (NLI) that translates user queries into structured Cypher queries for graph-based storage systems. The approach was implemented and evaluated with over 530 natural language queries on two publicly available event logs, demonstrating its effectiveness. This solution simplifies the process of data querying for analysts, making it more accessible and user-friendly.

4 Prompting Method

This chapter discusses a proposed prompt design for question-answering in process mining.

4.1 Prompt Overview

The prompt was separated into different elements to enhance the LLM's understanding of its content. A number of strategies that have proven to improve an LLM's performance were also employed.

First off, the persona of an expert in process mining can be assigned to the LLM. The first sentence in the prompt can read thus: You are an expert in process mining. This role is intended to ensure the model's responses align with the expert-level reasoning and analysis of a process analyst.

The performance statistics of the activities and transitions can be split into two tables: the activity table and the transition table. The activity table contains the performance statistics of all activities in the process, for instance, the case frequency of "Activity A." Whereas the transition table contains the performance statistics of all transitions in the process, the case frequency of "Task A → Task B" transition.

Apromore Process Discoverer is a tool that can help extract these performance statistics. The performance statistics extracted can be classified into three:

- Frequency metric: This includes the average, median, maximum, minimum, and total frequency of activity or transition. It also includes the case frequency of an activity or transition, which is the number of cases in which the activity or transition appeared.
- Duration metric: This includes the average, median, maximum, minimum, and total duration of an activity or transition. The duration is the time between the start and end of an activity or transition.
- Cost metric: Similar to the duration metric, it includes the average, median, maximum, minimum, and total cost of an activity or transition. The cost is determined by multiplying the duration of an activity instance by the unit cost of the corresponding resource performing that activity. This unit cost is typically inputted by the user into the process mining tool.

Since the LLM may fail to grasp specific process mining terms and would fall back to its generic interpretation, the prompt should have the definitions of key process mining terms. This includes the definition of all the performance statistics extracted and process mining terms such as activity, transition, bottleneck, rework, etc.

To further refine the LLM's output, the prompt should include specific instructions on how the LLM should structure its response. Specifically, the prompt can instruct the LLM to provide its chain of thoughts, detailing the logical steps leading to its final answer. This allows the LLM to reflect on the approach to answer the question accurately before proceeding to answer it. Additionally, relevant notes can be included to guide the LLM's output further. For instance, the LLM can be explicitly told to provide answers solely from the information in the prompt. Although an understanding of the prompt and question is required, the answer should not involve information outside the context of the process. A more detailed explanation of the prompt is discussed in sections 4.2.1 and 4.2.2.

4.2 Prompt Approaches

The prompts can be designed using two approaches. In the first approach, the prompt contained the performance statistics derived from the Apromore Process Discoverer. In the second approach, the performance statistics were not a part of the prompt but rather stored in an external relational database. Sections 4.2.1 and 4.2.1 dive deeper into the nuances between both approaches.

4.2.1 Direct Prompt Mode

This approach entails including all the process statistics, formatted as the text of a CSV file, directly into the prompt. The implication is a much larger prompt, especially as the process event data gets larger. The LLM gets the data about the process and is expected to parse and understand the data and extract the accurate answer to the question from the prompt.

The LLM is expected to rely only on its understanding of the information in the prompt. Therefore, its response should reflect insights drawn from the data in either the activity table or translation table. This approach will also test the LLM's capacity to apply process mining knowledge to a particular process. It is, however, advantageous as the LLM directly finds answers from the prompt without any further computation.

According to the prompting guideline as mentioned in section 2.4.2, the prompt was divided into five sections:

I. Assigning the role and goal of the task: The LLM was said to be an expert in process mining, and its goal is to answer specific process mining questions based on subsequent information in the prompt.
II. Passing the activity and transition performance statistics: The texts from the extracted performance statistics were passed. They were split into activity and transition tables.
III. Definition of terms: The terms related to performance metrics and process mining were defined.
IV. Notes: This involved some notes on how the questions should be answered. For instance, to get a precise answer, the LLM was asked to answer every question solely from information in the prompt.
V. Order of answer: To ensure the LLM displayed its chain of thoughts, it was asked to state its answer in a systematic way using three subheads: STEPS, DESCRIPTION, and ANSWER. The STEPS represents a high-level overview of how the LLM will arrive at its answer. The DESCRIPTION represents the implementation of the steps in relation to the data in the prompt. The ANSWER is the answer the LLM arrives at.

An example of this prompt approach is shown in Appendix 1.

4.2.2 Data Schema in the Prompt

In this approach, the activity table and transition table are stored in a relational database, not within the prompt. The LLM is provided with the data schema of the process statistics stored in a database. The relationships and types of data stored within the database are also passed to the LLM. With this information, the LLM is expected to return SQL queries that extract the required information to answer the question. The SQL queries are then executed on the database.

According to the prompting guideline as mentioned in section 2.4.2, the prompt was divided into five sections:

I. Assigning the role and goal of the task: The LLM was said to be an expert in process mining and a data engineer. Its goal is to answer specific process mining questions by returning the appropriate SQL query that returns the answer.

II. Passing the activity and transition schema: The schema of the activity and transition tables were passed here.

III. Definition of terms: The terms related to performance metrics and process mining were defined.

IV. Notes: This involved some notes on how the questions should be answered. For instance, in order to minimize errors in SQL queries, the LLM was asked not to make assumptions about column names.

V. Order of answer: To ensure the LLM displayed its chain of thoughts, it was asked to state its answer in a systematic way using three subheads: STEPS, DESCRIPTION, and ANSWER. The STEPS represents a high-level overview of how the LLM will arrive at its answer. The DESCRIPTION represents the implementation of the steps in relation to the database schema. The ANSWER is the SQL query that is used to arrive at the answer.

An example of this prompt approach is shown in Appendix 2.

4.3 Prompt Variations

To understand how LLMs react to changes in the prompting method, different prompt variations will be used. This is strictly for the direct prompt mode approach.

In the first variant, the prompt can have a more robust definition of terms. This is the upside prompt, while the baseline prompt has fewer definitions of terms. For example, the baseline prompts have the performance statistics of the nodes and transitions defined together (e.g., max duration of an activity/arc is the maximum duration of an activity or arc), whereas the upside prompt has them separately defined (e.g., activity max duration: Among all the durations of a specific activity in a log, this is the slowest duration. Transition max duration: Among all the durations of a specific transition in a log, this is the slowest duration). The upside prompt also can have new definitions, such as a log, a case, a transition bottleneck, an active bottleneck, etc. The baseline definition of terms is shown in Appendix 3, and the upside definition of terms is shown in Appendix 4.

In the second variation, the prompt sections can be compartmentalized with XML tags (<>, </>). The inclusion of these tags is inspired by the prompting guideline of Anthropic Claude[7]. These tags can be useful when the prompt is significantly larger, and the LLM does not pay attention to specific parts of the prompt.

[7] https://docs.anthropic.com/claude/docs/use-xml-tags

5 Experiment Setup

This section outlines the experimental setup, specifically the choice of the LLM, the process mining questions, the prompt design, and the criteria for evaluating model responses.

The experiment began by collecting the data sets and process mining questions that can be asked. Using the Apromore Process Discoverer, the ground truth for the questions was prepared. The LLM were asked the questions and returned their answers. The answers were compared to the ground truths as a measure of goodness. The recall, precision, F1-score, and accuracy were then prepared for evaluation. Figure 3 shows a rundown of the experimental procedure using the BPMN diagram.

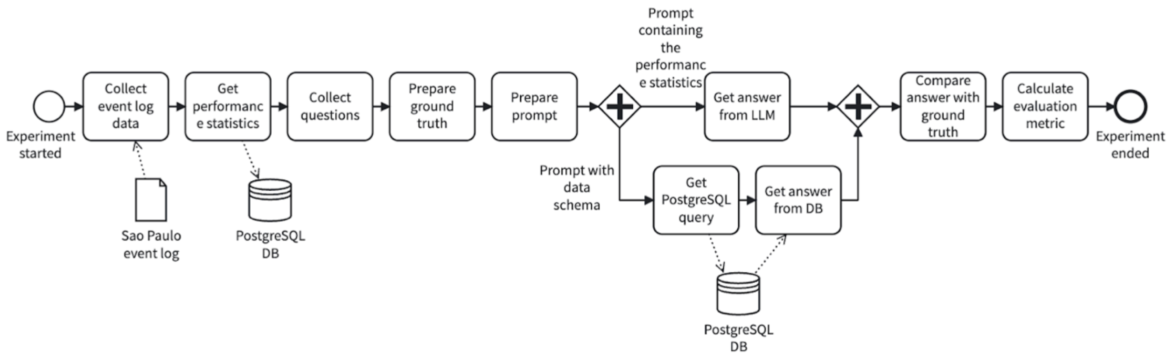

Figure 3: A BPMN diagram showing the experimental procedure.

The prompts were slightly varied to test different prompting strategies. In one variable, XML tags were added to different sections of the prompt. For instance, the paragraph containing the activity table started and ended with <activity_table> and </activity_table>, respectively. The XML tags added are: <task_definition>, <activity_table>, <transition_table>, <terms_defined>, </extra_notes> and <answer_format>.

Also, the terms of process mining were defined in two different ways. In the first variants, the definitions were shorter, perhaps not as descriptive. However, in the sections, the definitions were more descriptive, making them longer. More process mining terms were added in the second variant, making the prompt longer than in the first variant.

5.1 Choice of LLMs

To conduct the experiment, two state-of-the-art LLMs, GPT-4 and Claude V3 were used. These models have proven to have advanced question-answering capabilities and outperform other LLMs [50].

GPT-4 was used through the ChatGPT interface. As a paid user, ChatGPT used the latest `gpt-4-turbo-2024-04-09` model, with improved capabilities in logical reasoning, writing, and coding over the previous models. Claude V3 was accessed through the chat interface on Anthropic. Specifically, Claude 3 Opus, the most intelligent model from Anthropic, was used.

Both models were used in parallel between March 2024 and April 2024.

5.2 Combination of Questions

A set of 20 process mining questions focused on process discovery were selected. These were questions that business stakeholders ask about their process [41]. The authors proposed a list of 794 questions, which are categorized into event log data, analysis, process model, and advanced analysis. Questions from the categories require a specific abstraction of the event log. For the sake of narrowing down the scope of the experiment, questions from the event log data and process model were excluded. In addition, the performance statistics abstraction did not contain the event log or process model. Hence, we cannot answer questions in those categories. There were 327 questions in the event log category and 107 questions in the process model category. The analysis and advanced analysis categories had 240 and 120 questions, respectively. Of these, ten analyses focused on process discovery were selected, and eight advanced analysis questions focused on process optimization were selected. Two other questions related to the log nuances were selected. For example, how would this log behave if I remove DJE? Table 2 shows examples of the questions selected:

Table 2: Example of questions selected.

Category	Question
Analysis	What is the most frequent activity?
Analysis	What are the main causes of rework in this process?
Advanced analysis	How would my process behave if I removed task A?

The full list of questions is in Appendix 5.

5.3 Datasets

The dataset of interest needed to be a relatively complex dataset with numerous case variants, activity, and transition instances. To scope the experiment under a reasonable computational requirement, the dataset should, however, not be too long, with millions of activities and transitions. In addition, the dataset had to be publicly available. To prevent bias, the process also needed to be less popular to reduce the likelihood of the LLM being too conversant with the process. Amongst the datasets used in published literature, the Tribunal São Paulo event log fulfilled the established criteria.

The São Paulo Tribunal event log provides a comprehensive record of the activities and proceedings conducted by the Tribunal de Contas do Município de São Paulo (TCM-SP) [56]. The event log has 4600 case variants, 266,800 activity instances, and 4800 cases with 430 activities. The event log contains the proceedings from 19th December 2008 to 30th December 2020. The labels of the event log were in Portuguese but were translated to English using the Google Translate API.

Because the activity and transitions in the process are relatively large, the abstraction slider in Apromore was used to select a subset of the most frequent activities and transitions. The log was reduced to retain the following:

- The top 5% of the activities and 100% of transitions, called **SP-small**: The number of activities retained is small, leading to a smaller number of transitions. Hence, the entire transition left was selected.
- The top 8% of the activities and 25% of the transitions, called **SP-medium**: Increasing the number of nodes exponentially increases the number of transitions left. Hence, 25% of the top transitions were selected.
- The top 10% of the activities and 20% of the transitions, called **SP-large**: Beyond 10%, the process map explodes. Hence, the number of nodes was limited to the top 10%.

The same questions were asked on all three abstracted logs, each significantly larger in size than the previous. In other words, the size of SP-small < SP-medium < SP-large.

Table 3 shows the statistics of the three abstracted process maps.

Table 3: Statistics of the abstracted process maps

Statistics	**SP-small**	**SP-medium**	**SP-large**
Number of activities	8	15	22
Number of transitions	44	46	79
Number of words in the prompt	2360	2806	3962
Number of characters in the prompt	11830	13247	18311

It is important to note that when the user changes the abstraction settings in Apromore to a percentage, the number of nodes/transitions may not be entirely the percentage of the total number of nodes/transitions. Apromore ensures that every node is on a path from the start node to the end node. For example, if there are 50 total nodes and a 10% node abstraction setting is used, Apromore checks that the five nodes to be retained are on a path from the start node to the end node. If one node is not in that path, that node will not be displayed. Thus, four nodes will be retained for a 10% abstraction setting with 50 total nodes. Due to these heuristics, a percentage abstraction slider should be considered as an actual percentage calculation.

5.4 Evaluation Criteria

The responses generated by the LLMs were evaluated based on a scoring framework designed to assess the precision and recall of the models' responses. The ground truth was self-determined based on the expected result from the process mining tool, Apromore. The responses from the LLM were then evaluated based on this ground truth to determine its measure of goodness. The LLM evaluation criteria were different for the two prompting approaches used.

In the direct prompt mode, the LLM may be required to return multiple answers. Hence, the evaluation criteria classify responses into three levels: incorrect, partially correct, and correct. Each category is assigned a single point value for correctness and completeness.

A response is classified as "incorrect" when it does not address the question asked or is fundamentally wrong. For instance, if the LLM is asked to state the source of the bottleneck in the process and it states an activity is not a bottleneck, the answer is incorrect. Incorrect responses contribute zero points to the overall score. Responses that contain correct answers but are incomplete are classified as "partially correct." In this case, the concepts of true positives (TP), false positives (FP), and false negatives (FN) were used to determine the recall, precision, and F1 score.

- True Positives (TP): The answers that are correctly identified.
- False Positives (FP): The answers that are incorrectly identified.
- False Negatives (FN): The correct answers that are not identified.

With these, the precision, recall, and F1-score are given by equations i, ii, and iii, respectively.

$$Precision = \frac{TP}{TP + FP} \qquad \text{(i)}$$

$$Recall = \frac{TP}{TP + FN} \qquad \text{(ii)}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad \text{(iii)}$$

For instance, the question may be asked to state the bottlenecks in the process. If there are two bottlenecks, but the LLM stated just one, the response is partially correct. The precision is one while the recall ½. A response is considered “correct” when it accurately and completely addresses the question posed. For instance, if the LLM is asked to state the bottlenecks in the process and there are two bottlenecks, the LLM must state both bottlenecks. Correct responses are awarded 1 point.

The total score for each LLM is calculated by summing the points awarded across all process mining questions. This scoring system would help perform a quantitative comparison between both LLMs.

On the other hand, for the prompt with the database schema, the LLM is expected to return a valid SQL query. The query can either be correct or incorrect. Hence, the evaluation criteria were based on accuracy. An answer accuracy attracts one point if the query is syntactically correct and returns the expected answer. If the query is not syntactically correct or returns a wrong answer, this means the accuracy of the question is zero.

6 Result and Discussion

In this chapter, the performance of GPT-4 and Claude V3 is assessed. The assessments were separately done for both prompting approaches.

6.1 Direct Answering

In this approach, the prompt contained information about the process as well as the context and the answer format. Both GPT-4 and Claude V3 returned the answer in the expected format, indicating an initial understanding of the prompt. The prompt was tested on SP-small, SP-medium, and SP-large to understand whether an increase in the number of activities and transitions affected the LLM's performance.

Furthermore, the prompt was varied to have a more explicit but lengthier definition of terms. Finally, the effect of adding XML tags was investigated.

6.1.1 Variation of Event Log Size

Here, the LLM performance was evaluated for SP-small, SP-medium, and SP-large.

For example, both were asked, "What activity is the major bottleneck in the process?".

GPT-4 response:

STEPS:

Review the "Activity" table and identify the activity with the highest "Total duration."

Compare this with the "Transition" table to ensure there's no transition leading to an activity with a higher average duration that might indicate a bottleneck at the activity level.

Confirm the activity with the highest "Total duration" as the major bottleneck if no other activity has a higher average duration caused by a transition.

DESCRIPTION:

To identify the major bottleneck in the process, one must consider both the "Activity" and "Transition" tables. The major bottleneck at the activity level is the one with the highest "Total duration," indicating it takes the longest time to complete. However, it's essential to also examine the transitions leading to activities, as a transition with a high average duration could indicate a bottleneck in the flow towards an activity, even if the activity itself does not have the highest total duration. The bottleneck is thus determined by a combination of total and average durations across activities and transitions, identifying where delays are most significant.

ANSWER:

The major bottleneck in the process, based on the "Activity" table, is DJE, with a "Total duration" of 3.85 years. This activity takes the longest time to complete, indicating it is a significant source of delay in the process.

Claude V3 response:

STEPS:

1. Examine the "Average duration" column in the Activity table to identify activities with relatively high average durations, as these could be potential bottlenecks.

2. Look at the "Total duration" column in the Transition table and identify the transition(s) with the highest total duration. The target activity of this transition is likely to be a major bottleneck.

3. Cross-check the findings from steps 1 and 2 to determine the most significant bottleneck activity.

DESCRIPTION:

Bottlenecks in a process can be identified by analyzing activities with relatively long execution times or transitions that take a long time to complete. In the Activity table, activities with high "Average duration" values could indicate potential bottlenecks. However, the Transition table provides more direct insights into bottlenecks, as the transition with the highest "Total duration" suggests that the target activity is struggling to start promptly after the source activity is completed. By combining the findings from both tables, we can determine the most significant bottleneck activity in the process.

ANSWER:

From the Activity table, the activity with the highest "Average duration" is "Petition" at 1.06 hours, indicating a potential bottleneck.

However, in the Transition table, the transition from "DJE" to "Publication Certificate issued" has the highest "Total duration" of 115.9 years. This suggests that the target activity, "Publication Certificate issued," is a major bottleneck, as it takes a long time to start after the completion of the "DJE" activity.

Table 4 shows the performance of GPT-4 and Claude V3. The highest metrics are written in **bold** text.

Table 4: GPT-4 and Claude V3 performance on the correctness and completeness of its answers.

LLM / Log data	**GPT-4**			**Claude V3**		
	Recall (%)	**Precision (%)**	**F1-score (%)**	**Recall (%)**	**Precision (%)**	**F1-score (%)**
SP-small	72.33	71.33	71.83	**80.00**	**75.28**	**77.56**
SP-medium	70.00	67.25	68.60	**86.90**	**84.17**	**85.51**
SP-large	**76.43**	69.64	72.88	75.35	**81.50**	**78.30**

We observe that, for GPT-4, there is a decrease in recall and precision between SP-small and SP-medium. Claude V3, on the other hand, showed an increase in recall and precision from SP-small to SP-medium. This could indicate that Claude V3 can better deal with larger prompts than GPT-4. As the prompt got larger, however (as seen in SP-large), both GPT-4 and Claude V3 experienced a drop in performance.

It was observed, for instance, that for larger prompts, both LLMs struggled to give correct and complete answers. For instance, in the question, list all the self-loops in the process. In SP-medium and SP-large, there are four self-loops from Apromore Process Discoverer:

- Petition

- DJE
- Suspension of the term
- Positive air

Claude V3 listed five self-loops, with one being incorrect.

- "SUSPENSION OF THE TERM" to "SUSPENSION OF THE TERM"
- "Petition" to "Petition"
- "Positive air" to "Positive air"
- "DJE" to "DJE"
- "Decision" to "Decision" (incorrect)

GPT-4 listed three self-loops with one correct answer missing.

- "Positive air" to "Positive air"
- "DJE" to "DJE"
- "SUSPENSION OF THE TERM" to "SUSPENSION OF THE TERM"

In SP-large, however, Claude V3 listed all four self-loops correctly, while GPT-4 listed three self-loops with one correct answer missing.

6.1.2 Variation in definition of terms

Here, the definitions of terms were varied. The definition of terms changed to longer and more robust explanations. The result of the performance is shown in Table 5.

Table 5: LLMs performance with a different variation of terms.

LLM / Log data	**GPT-4**			**Claude V3**		
	Recall (%)	**Precision (%)**	**F1 score (%)**	**Recall (%)**	**Precision (%)**	**F1-score (%)**
SP-small	76.67	77.50	77.08	**82.50**	**80.67**	**81.57**
SP-medium	76.17	84	79.90	**87.33**	**86.07**	**86.69**
SP-large	69.46	**75.00**	**72.12**	**70.00**	71.19	70.60

Comparing this to the results of the vanilla prompt in Table 4, it can be observed that both LLMs perform better for SP-small but have a drop in performance for SP-medium and SP-large. This indicates that while better definitions helped the smaller prompt better, it further increased the prompt size of the larger prompt, tanking its performance.

Another observation was that GPT-4 had more precision than recall. In other words, GPT-4 would rather return a correct but incomplete answer — more like playing it safe. Claude V3, on the other hand, had slightly higher recall than precision. This is based on our observation that Claude V3 returns completely correct answers at the expense of having a few incorrect answers in the mix.

For instance, the question is, what other bottleneck exists in the process?

Ground truth from Apromore:

Activity bottleneck: Decision - average duration 1.86 hours

Transition bottleneck:

Freely distributed (by draw) (exclusive movement of the distributor) - average duration 1.91 hrs

Publication Certificate issued to Suspension of the term - average duration 2.98 weeks

Conclusions for Order to Decision - average duration 2.71 weeks

Conclusions for Order to Conclusions for decision - average duration 2.4 weeks

Suspension of the term to Conclusions for decision - average duration 2.57 weeks

GPT-4 answer:

Activity Bottlenecks:

"Decision" with an average duration of 1.86 hours.

"Petition" with an average duration of 1.06 hours.

"SUSPENSION OF THE TERM" with an average duration of 1.02 hours.

Transition Bottlenecks:

The transition with the highest average duration is "Publication Certificate issued, SUSPENSION OF THE TERM" with an average duration of 2.99 weeks. This transition indicates a significant waiting time, contributing to delays in the process flow.

These identified bottlenecks highlight areas where the process can potentially be optimized to improve overall efficiency.

Claude V3 answer:

Other bottlenecks that exist in the process are:

From the Activity table:

Decision: Average duration of 1.86 hrs

Positive air: Average duration of 1.29 hrs

Petition: Average duration of 1.06 hrs

From the Transition table:

Transition from "Publication Certificate issued" to "Ordinary Act - Publishable": Average duration of 2.22 weeks

Transition from "Conclusions for Order" to "Decision": Average duration of 2.71 weeks

Transition from "SUSPENSION OF THE TERM" to "SUSPENSION OF THE TERM" (self-loop): Average duration of 2.96 weeks

Transition from "SUSPENSION OF THE TERM" to "Conclusions for decision": Average duration of 2.57 weeks

These activities and transitions have relatively high average durations and could be considered as other bottlenecks in the process.

GPT-4 identified only one transition rework of four, playing it safe. Claude, V3, on the other hand, identified three activity transitions when, in fact, there is only one.

6.1.3 Adding XML tags

The results from adding XML tags were added to different segments of the prompts are presented. Table 6 shows the performance of XML tags. Table 6 shows the results.

Table 6: LLMs performance with XML tags in the prompt.

LLM / Prompt	GPT-4			Claude V3		
	Recall (%)	**Precision (%)**	**F1 score (%)**	**Recall (%)**	**Precision (%)**	**F1-score (%)**
SP-small	72.83	75.83	74.30	**77.50**	**77.33**	**77.41**
SP-medium	73.11	74.57	73.78	**76.18**	**75.33**	**75.72**
SP-large	72.92	72.62	72.77	**77.56**	**79.64**	**78.56**

It can be observed that the recall, precision, and F1-score for GPT-4 and Claude V3 remained largely the same with XML tags. It can be said that the result is inconclusive.

6.2 Returning SQL Queries

In this approach, GPT-4 and Claude V3 were asked to return an SQL query.

For the question, "What activity is the major bottleneck in the process?".

GPT-4 response:

```
-- Bottleneck from transitions table
SELECT Target_Node AS Bottleneck_Activity
FROM transitions
ORDER BY Average_duration DESC
LIMIT 1;
```

Claude V3 response:

```
-- Find the transition with highest average duration
SELECT Target_Node AS Bottleneck_Activity
FROM transitions
ORDER BY Average_duration DESC
LIMIT 1;
```

Table 7 shows the performance of GPT-4 and Claude V3 performance from the SQL query result.

Table 7: GPT-4 and Claude V3 performance on the accuracy of its answers

GPT-4	**Claude V3**
Accuracy (%)	Accuracy (%)

75.00	60.00

We see that GPT-4 performed better than Claude V3, indicating that GPT-4 was better at generating codes. Beyond just generating correct answers, GPT-4 codes appeared potentially more efficient. Furthermore, GPT-4 codes returned more correct answers than Claude V3, as seen from the accuracy score in Table 7. For example, in the question, what activities occurred in all cases? The following are GPT-4 and Claude V3 responses.

GPT-4 response:

```
SELECT Node AS Activities_In_All_Cases
FROM nodes
WHERE Case_frequency = (SELECT MAX(Case_frequency) FROM nodes);
```

Code output:

```
"activities_in_all_cases"
"Publication Certificate issued"
"DJE"
```

Claude V3 response:

```
SELECT Node
FROM nodes
WHERE Node NOT IN ('Start', 'End')
AND Case_frequency = (
SELECT Case_frequency
FROM nodes
E Node = 'Start'
);
```

Code output:

```
node
```

GPT-4 output is correct, while Claude's V3 code returns nothing.

In some other situations, Claude V3 did not properly render its code, leading to syntax errors. For the question, identify the rework loops in the process, Figure 4 showed a GPT-4 response while Figure 4 showed a Claude V3 response.

SUGGESTED QUERIES:

To identify rework loops:

```sql
SELECT Source_Node, Target_Node, Total_Rework_frequency, Total_Rework_duration
FROM transitions
WHERE Total_Rework_frequency > 0
ORDER BY Total_Rework_frequency DESC;
```

To identify high rework activities:

```sql
SELECT Node, Total_Rework_frequency, Total_Rework_duration
FROM nodes
WHERE Total_Rework_frequency > 0
ORDER BY Total_Rework_frequency DESC;
```

Figure 4: GPT-4 response to determining the rework loops.

ANSWER:
WITH RECURSIVE rework_paths AS (
SELECT Source_Node, Target_Node, ARRAY[Source_Node, Target_Node] AS path
FROM transitions

```
UNION ALL

SELECT rp.Source_Node, t.Target_Node, rp.path || t.Target_Node
FROM rework_paths rp
JOIN transitions t ON rp.Target_Node = t.Source_Node
WHERE rp.Source_Node <> t.Target_Node AND rp.Source_Node = ANY(rp.path[2:])
```

)
SELECT DISTINCT Source_Node, Target_Node
FROM rework_paths
WHERE array_length(path, 1) > 2;

Figure 5: Claude V3 response to determining the rework loops.

In this example, GPT-4 code output was correct, while that of Claude V3 was incorrect.

7 Conclusion and Future Work

The findings suggest that LLMs can perform process mining tasks, given that the prompt is not very large. The optimum prompt size cannot be quantitatively ascertained and can be an idea for other research. In this work, however, SP-medium gave a better result than SP-large, indicating a prompt size of 13247 characters (~3317 tokens) is better than a prompt size of 18311 characters (~4577 tokens).

The experiment showed that Claude V3 outperforms GPT-4 in answering process mining questions using the direct prompt approach. In cases where the context window is large, the LLM (particularly GPT-4) does not pay attention to specific parts of the prompt. Nevertheless, Claude V3 particularly showed that it could sieve through larger prompts to provide its answers correctly than its GPT-4 counterpart. It was also observed that GPT-4 had higher precision than recall, indicating that it would give incomplete answers than incorrect answers. Claude V3, on the other hand, gave a richer list of answers, although they may contain a few incorrect answers. With the RAG approach, however, where the LLMs are to return an SQL query, GPT-4 outperforms Claude V3.

Juxtaposing between both approaches, direct prompting seemed to be the better approach when the process was not large. It returns the answer faster with less computational power. The RAG approach, however, is preferable for larger prompts. Since the prompt only contains the database schema and not the process mining data itself, the SQL query the LLM generates does not change when the data in the database is large or small.

This study revealed that having the definition of domain-specific terms is crucial to improve the model's performance. Definitions such as a case, a log, a bottleneck, a rework, etc, should be added to the prompt. The use of XML tags in the prompt, however, did not yield any substantial improvement.

While this study was experimental, further work can be done to implement the RAG approach using tools such as LangChain[8]. The LLM query output can be connected to the database, and the result of the query is passed back to the LLM. For every question, an agent can be used to determine the necessary information the LLM needs to query from the database. It generates information from the database to answer questions. This allows the model to retrieve the requisite insight to answer nuanced questions about the process.

In addition, out-of-the-box LLMs can be further trained on process mining data. The pre-trained model can be fine-tuned for datasets specific to process mining. This can help the LLM understand the nuances of process mining, have domain-specific knowledge could potentially reduce misinterpretations and errors in its answers.

[8] https://www.langchain.com/

References

[1] I. Kaniški and I. Vincek, 'Business processes as business systems', *Teh. Glas.*, vol. 12, pp. 55–61, Mar. 2018, doi: 10.31803/tg-20170808183458.

[2] T. Sonteya and L. Seymour, 'Towards an Understanding of the Business Process Analyst: An Analysis of Competencies', *J. Inf. Technol. Educ.*, vol. 11, Jan. 2012, doi: 10.28945/1568.

[3] H. A. Reijers, 'Business Process Management: The evolution of a discipline', *Comput. Ind.*, vol. 126, p. 103404, Apr. 2021, doi: 10.1016/j.compind.2021.103404.

[4] W. Aalst *et al.*, 'Process Mining Manifesto', presented at the Lecture Notes in Business Information Processing, Aug. 2011, pp. 169–194. doi: 10.1007/978-3-642-28108-2_19.

[5] P. Badakhshan, B. Wurm, T. Grisold, J. Geyer-Klingeberg, J. Mendling, and J. vom Brocke, 'Creating Business Value with Process Mining', *J. Strateg. Inf. Syst.*, vol. 31, p. 2023, Oct. 2022, doi: 10.1016/j.jsis.2022.101745.

[6] I. Beerepoot *et al.*, 'The biggest business process management problems to solve before we die', *Comput. Ind.*, vol. 146, p. 103837, Apr. 2023, doi: 10.1016/j.compind.2022.103837.

[7] S. van Zelst, 'Event abstraction in process mining: literature review and taxonomy', *Granul. Comput.*, Jan. 2020, Accessed: Mar. 14, 2024. [Online]. Available: https://www.academia.edu/87553719/Event_abstraction_in_process_mining_literature_review_and_taxonomy

[8] W. M. P. van der Aalst, 'Analyzing "Spaghetti Processes"', in *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, W. M. P. van der Aalst, Ed., Berlin, Heidelberg: Springer, 2011, pp. 301–317. doi: 10.1007/978-3-642-19345-3_12.

[9] L. Zimmermann, F. Zerbato, and B. Weber, 'What makes life for process mining analysts difficult? A reflection of challenges', *Softw. Syst. Model.*, Nov. 2023, doi: 10.1007/s10270-023-01134-0.

[10] A. Berti and M. Qafari, *Leveraging Large Language Models (LLMs) for Process Mining (Technical Report)*. 2023.

[11] E. Kamalloo, N. Dziri, C. Clarke, and D. Rafiei, 'Evaluating Open-Domain Question Answering in the Era of Large Language Models', in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 5591–5606. doi: 10.18653/v1/2023.acl-long.307.

[12] J. Kim *et al.*, 'SuRe: Improving Open-domain Question Answering of LLMs via Summarized Retrieval', presented at the The Twelfth International Conference on Learning Representations, Oct. 2023. Accessed: Feb. 29, 2024. [Online]. Available: https://openreview.net/forum?id=w4DW6qkRmt

[13] Z. Rasool *et al.*, 'Evaluating LLMs on Document-Based QA: Exact Answer Selection and Numerical Extraction using Cogtale dataset'. arXiv, Jan. 03, 2024. Accessed: Feb. 29, 2024. [Online]. Available: http://arxiv.org/abs/2311.07878

[14] H. Naveed *et al.*, 'A Comprehensive Overview of Large Language Models'. arXiv, Feb. 20, 2024. Accessed: Feb. 29, 2024. [Online]. Available: http://arxiv.org/abs/2307.06435

[15] '[2311.12785] Prompting Frameworks for Large Language Models: A Survey'. Accessed: May 15, 2024. [Online]. Available: https://arxiv.org/abs/2311.12785

[16] ‘Positioning and Presenting Design Science Research for Maximum Impact’. Accessed: Apr. 11, 2024. [Online]. Available: https://misq.umn.edu/positioning-and-presenting-design-science-research-for-maximum-impact.html
[17] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, ‘Introduction to Business Process Management’, in *Fundamentals of Business Process Management*, M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, Eds., Berlin, Heidelberg: Springer, 2018, pp. 1–33. doi: 10.1007/978-3-662-56509-4_1.
[18] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer, ‘Enterprise Data Analysis and Visualization: An Interview Study’, *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2917–2926, Dec. 2012, doi: 10.1109/TVCG.2012.219.
[19] K. Kaushal, ‘Social desirability bias in face to face interviews’, *J. Postgrad. Med.*, vol. 60, pp. 415–6, Oct. 2014, doi: 10.4103/0022-3859.143989.
[20] J. vom Brocke, M. Jans, J. Mendling, and H. A. Reijers, ‘A Five-Level Framework for Research on Process Mining’, *Bus. Inf. Syst. Eng.*, vol. 63, no. 5, pp. 483–490, Oct. 2021, doi: 10.1007/s12599-021-00718-8.
[21] A. Augusto *et al.*, ‘Automated Discovery of Process Models from Event Logs: Review and Benchmark’. arXiv, Jan. 29, 2018. Accessed: Mar. 07, 2024. [Online]. Available: http://arxiv.org/abs/1705.02288
[22] W. M. P. van der Aalst, ‘Process Mining: A 360 Degree Overview’, in *Process Mining Handbook*, W. M. P. van der Aalst and J. Carmona, Eds., in Lecture Notes in Business Information Processing. , Cham: Springer International Publishing, 2022, pp. 3–34. doi: 10.1007/978-3-031-08848-3_1.
[23] A. Rozinat and W. M. P. van der Aalst, ‘Conformance checking of processes based on monitoring real behavior’, *Inf. Syst.*, vol. 33, no. 1, pp. 64–95, Mar. 2008, doi: 10.1016/j.is.2007.07.001.
[24] F. Taymouri, M. L. Rosa, M. Dumas, and F. M. Maggi, ‘Business process variant analysis: Survey and classification’, *Knowl.-Based Syst.*, vol. 211, p. 106557, Jan. 2021, doi: 10.1016/j.knosys.2020.106557.
[25] C. dos S. Garcia *et al.*, ‘Process mining techniques and applications – A systematic mapping study’, *Expert Syst. Appl.*, vol. 133, pp. 260–295, Nov. 2019, doi: 10.1016/j.eswa.2019.05.003.
[26] Z. Ji *et al.*, ‘Survey of Hallucination in Natural Language Generation’, *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, Dec. 2023, doi: 10.1145/3571730.
[27] T. Winograd, ‘Understanding natural language’, *Cognit. Psychol.*, vol. 3, no. 1, pp. 1–191, Jan. 1972, doi: 10.1016/0010-0285(72)90002-3.
[28] V. H. Yngve, ‘A programming language for mechanical translation’.
[29] D. Khurana, A. Koli, K. Khatter, and S. Singh, ‘Natural language processing: state of the art, current trends and challenges’, *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.
[30] A. Vaswani *et al.*, ‘Attention Is All You Need’. arXiv, Aug. 01, 2023. doi: 10.48550/arXiv.1706.03762.
[31] A. D. Cheok and E. Y. Zhang, ‘From Turing to Transformers: A Comprehensive Review and Tutorial on the Evolution and Applications of Generative Transformer Models’, *Qeios*, Oct. 2023, doi: 10.32388/3NTOLQ.2.
[32] M. Raiaan *et al.*, *A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges*. 2023. doi: 10.36227/techrxiv.24171183.
[33] T. Eloundou, S. Manning, P. Mishkin, and D. Rock, ‘GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models’. arXiv, Aug. 21, 2023. Accessed: May 15, 2024. [Online]. Available: http://arxiv.org/abs/2303.10130

[34] Z. Xu, S. Jain, and M. Kankanhalli, ‘Hallucination is Inevitable: An Innate Limitation of Large Language Models’. arXiv, Jan. 22, 2024. doi: 10.48550/arXiv.2401.11817.
[35] A. Ayaz, A. Nawalgaria, and R. Yin, ‘Taught by the Internet, Exploring Bias in OpenAIs GPT3’. arXiv, Jun. 04, 2023. doi: 10.48550/arXiv.2306.02428.
[36] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, ‘Universal and Transferable Adversarial Attacks on Aligned Language Models’. arXiv, Dec. 20, 2023. doi: 10.48550/arXiv.2307.15043.
[37] X. Xu *et al.*, ‘An LLM can Fool Itself: A Prompt-Based Adversarial Attack’. arXiv, Oct. 20, 2023. doi: 10.48550/arXiv.2310.13345.
[38] G. Marvin, N. Hellen Raudha, D. Jjingo, and J. Nakatumba-Nabende, ‘Prompt Engineering in Large Language Models’, 2024, pp. 387–402. doi: 10.1007/978-981-99-7962-2_30.
[39] S. K. K. Santu and D. Feng, ‘TELeR: A General Taxonomy of LLM Prompts for Benchmarking Complex Tasks’. arXiv, Oct. 24, 2023. Accessed: Apr. 13, 2024. [Online]. Available: http://arxiv.org/abs/2305.11430
[40] P. Ceravolo, S. Barbon Junior, E. Damiani, and W. Aalst, *Tailoring Machine Learning for Process Mining*. 2023.
[41] U. Jessen, M. Sroka, and D. Fahland, ‘Chit-Chat or Deep Talk: Prompt Engineering for Process Mining’. arXiv, Jul. 19, 2023. doi: 10.48550/arXiv.2307.09909.
[42] A. Kong *et al.*, ‘Better Zero-Shot Reasoning with Role-Play Prompting’. arXiv, Mar. 13, 2024. doi: 10.48550/arXiv.2308.07702.
[43] J. Wei *et al.*, ‘Chain-of-Thought Prompting Elicits Reasoning in Large Language Models’. arXiv, Jan. 10, 2023. doi: 10.48550/arXiv.2201.11903.
[44] ‘Use XML tags’, Claude. Accessed: Mar. 28, 2024. [Online]. Available: https://docs.anthropic.com/claude/docs/use-xml-tags
[45] Y. Gao *et al.*, ‘Retrieval-Augmented Generation for Large Language Models: A Survey’. arXiv, Mar. 27, 2024. doi: 10.48550/arXiv.2312.10997.
[46] M. Grohs, L. Abb, N. Elsayed, and J.-R. Rehse, ‘Large Language Models can accomplish Business Process Management Tasks’. arXiv, Jul. 19, 2023. Accessed: Dec. 06, 2023. [Online]. Available: http://arxiv.org/abs/2307.09923
[47] M. Vidgof, S. Bachhofner, and J. Mendling, ‘Large Language Models for Business Process Management: Opportunities and Challenges’. arXiv, Apr. 09, 2023. Accessed: Feb. 29, 2024. [Online]. Available: http://arxiv.org/abs/2304.04309
[48] Z. Zheng *et al.*, ‘A Survey of Large Language Models for Code: Evolution, Benchmarking, and Future Trends’. arXiv, Jan. 08, 2024. doi: 10.48550/arXiv.2311.10372.
[49] ‘ChatGPT — Release Notes | OpenAI Help Center’. Accessed: Apr. 11, 2024. [Online]. Available: https://help.openai.com/en/articles/6825453-chatgpt-release-notes
[50] ‘Introducing the next generation of Claude’. Accessed: Mar. 21, 2024. [Online]. Available: https://www.anthropic.com/news/claude-3-family
[51] A. Berti, H. Kourani, H. Hafke, C.-Y. Li, and D. Schuster, ‘Evaluating Large Language Models in Process Mining: Capabilities, Benchmarks, Evaluation Strategies, and Future Challenges’. arXiv, Mar. 11, 2024. doi: 10.48550/arXiv.2403.06749.
[52] A. Berti, D. Schuster, and W. M. P. van der Aalst, ‘Abstractions, Scenarios, and Prompt Definitions for Process Mining with LLMs: A Case Study’. arXiv, Jul. 14, 2023. Accessed: Dec. 06, 2023. [Online]. Available: http://arxiv.org/abs/2307.02194
[53] L. Barbieri, E. R. M. Madeira, K. Stroeh, and W. M. P. Van Der Aalst, ‘Towards a Natural Language Conversational Interface for Process Mining’, in *Process Mining Workshops*, vol. 433, J. Munoz-Gama and X. Lu, Eds., in Lecture Notes in Business Information Processing, vol. 433. , Cham: Springer International Publishing, 2022, pp. 268–280. doi: 10.1007/978-3-030-98581-3_20.

[54] C. Kecht, A. Egger, W. Kratsch, and M. Röglinger, ‘Quantifying chatbots’ ability to learn business processes’, *Inf. Syst.*, vol. 113, p. 102176, Jan. 2023, doi: 10.1016/j.is.2023.102176.
[55] M. Kobeissi, N. Assy, W. Gaaloul, B. Defude, B. Benatallah, and B. Haidar, ‘Natural language querying of process execution data’, *Inf. Syst.*, vol. 116, no. C, Jun. 2023, doi: 10.1016/j.is.2023.102227.
[56] A. Borji and M. Mohammadian, ‘Battle of the Wordsmiths: Comparing ChatGPT, GPT-4, Claude, and Bard’. Rochester, NY, Jun. 12, 2023. doi: 10.2139/ssrn.4476855.

Appendix

I. Sample prompt with performance statistics

You are an expert in process mining. I want you to answer some questions about the performance of a process.

The following table gives you the performance statistics for every activity in an event log. It is a CSV with , delimiter

Node,Min duration,Total Rework frequency,Average duration,Max duration,Min frequency,Case frequency,Average frequency,Total Rework duration,Max frequency,Total frequency,Median duration,Median frequency,Total duration,Relative case frequency

End,0,0,0,0,0,0,0,0,0,0,0,0,0,0

Freely distributed (by draw) (exclusive movement of the distributor),instant,0,1.91 hrs,2.43 mths,0,"4,123",0.86,instant,1,"4,123",instant,1,10.78 mths,85.99%

DJE,instant,"33,072",53.42 mins,4.04 mths,1,"4,795",7.9,3.49 yrs,50,"37,867",instant,7,3.85 yrs,100%

Start,0,0,0,0,0,0,0,0,0,0,0,0,0,0

Petition,instant,"21,233",1.06 hrs,2.73 mths,0,"4,187",5.3,2.69 yrs,68,"25,420",instant,3,3.09 yrs,87.32%

Publication Certificate issued,instant,"32,059",26.77 mins,1.29 wks,1,"4,795",7.69,1.6 yrs,47,"36,854",instant,7,1.88 yrs,100%

Registry certificate issued,instant,"6,284",39.89 mins,2.17 mths,0,"4,149",2.18,4.9 mths,17,"10,433",instant,2,9.49 mths,86.53%

Conclusions for decision,instant,"25,504",51.35 mins,11.89 mths,0,"4,789",6.32,1.73 yrs,41,"30,293",instant,5,2.96 yrs,99.87%

The following table gives you for every transition between two activities that occur in an event log. This is a CSV with , delimiter

Source Node,Target Node,Min duration,Total Rework frequency,Average duration,Max duration,Min frequency,Case frequency,Average frequency,Total Rework duration,Max frequency,Total frequency,Median duration,Median frequency,Total duration,Relative case frequency

Petition,DJE,instant,738,2.43 days,4.93 mths,0,"1,574",0.48,4.98 yrs,10,"2,312",1 day,0,15.4 yrs,32.83%

Freely distributed (by draw) (exclusive movement of the distributor),Publication Certificate issued,1 day,0,2.93 days,6 days,0,14,0,instant,1,14,2 days,0,1.35 mths,0.29%

Start,Conclusions for decision,instant,0,instant,instant,0,762,0.16,instant,1,762,instant,0,instant,15.89%

Publication Certificate issued,End,instant,0,instant,instant,0,920,0.19,instant,1,920,instant,0,instant,19.19%

Registry certificate issued,Conclusions for decision,instant,62,5.62 days,5.32 mths,0,"1,014",0.22,1.98 yrs,3,"1,076",instant,0,16.55 yrs,21.15%

Publication Certificate issued,Petition,instant,"5,367",1.47 wks,1.22 yrs,0,"3,324",1.81,155.94 yrs,18,"8,691",1 wk,1,244.44 yrs,69.32%

Petition,Conclusions for decision,instant,"1,901",1.09 wks,6.7 mths,0,"2,384",0.89,43.15 yrs,11,"4,285",3 days,0,89.16 yrs,49.72%

Publication Certificate issued,DJE,instant,342,2.03 days,3.94 mths,0,"1,060",0.29,1.51 yrs,9,"1,402",1 day,0,7.77 yrs,22.11%

DJE,Registry certificate issued,instant,16,3.4 days,1.08 mths,0,280,0.06,1.08 mths,2,296,1 day,0,2.75 yrs,5.84%

Freely distributed (by draw) (exclusive movement of the distributor),Conclusions for decision,instant,0,3.35 days,2.27 mths,0,839,0.17,instant,1,839,1 day,0,7.7 yrs,17.5%

Conclusions for decision,Publication Certificate issued,instant,79,14.51 hrs,2.71 wks,0,500,0.12,3.43 wks,6,579,instant,0,11.5 mths,10.43%

Conclusions for decision,Petition,instant,"1,353",3.21 days,4.17 mths,0,"1,746",0.65,14.85 yrs,10,"3,099",instant,0,27.24 yrs,36.41%

Conclusions for decision,Registry certificate issued,instant,954,3.94 hrs,2.03 mths,0,"2,811",0.79,8.27 mths,7,"3,765",instant,1,1.69 yrs,58.62%

Registry certificate issued,Publication Certificate issued,instant,42,9.91 hrs,1.14 wks,0,561,0.13,2.29 wks,2,603,instant,0,8.18 mths,11.7%

Registry certificate issued,End,instant,0,instant,instant,0,419,0.09,instant,1,419,instant,0,instant,8.74%

Conclusions for decision,Freely distributed (by draw) (exclusive movement of the distributor),instant,0,instant,instant,0,732,0.15,instant,1,732,instant,0,instant,15.27%

Publication Certificate issued,Conclusions for decision,instant,"1,411",1.84 wks,10.68 mths,0,"2,107",0.73,53.26 yrs,9,"3,518",1.01 wks,0,124.19 yrs,43.94%

Start,Freely distributed (by draw) (exclusive movement of the distributor),instant,0,instant,instant,0,"2,509",0.52,instant,1,"2,509",instant,1,instant,52.33%

Freely distributed (by draw) (exclusive movement of the distributor),Petition,instant,0,2.49 days,4.14 wks,0,200,0.04,instant,1,200,1 day,0,1.36 yrs,4.17%

Registry certificate issued,Petition,instant,131,1.27 wks,1.23 yrs,0,767,0.19,2.27 yrs,4,898,1 day,0,21.85 yrs,16%

Petition,Freely distributed (by draw) (exclusive movement of the distributor),instant,0,instant,instant,0,55,0.01,instant,1,55,instant,0,instant,1.15%

Petition,End,instant,0,instant,instant,0,481,0.1,instant,1,481,instant,0,instant,10.03%

Start,Registry certificate issued,instant,0,instant,instant,0,72,0.02,instant,1,72,instant,0,instant,1.5%

Conclusions for decision,DJE,instant,"2,828",2.57 days,4.6 mths,0,"3,089",1.23,20.62 yrs,16,"5,917",1 day,1,41.64 yrs,64.42%

DJE,Conclusions for decision,instant,111,1.56 days,2.86 wks,0,620,0.15,5 mths,7,731,1 day,0,3.13 yrs,12.93%

Start,DJE,instant,0,instant,instant,0,18,0,instant,1,18,instant,0,instant,0.38%

Registry certificate issued,DJE,instant,287,2.71 days,5.19 mths,0,"1,423",0.36,1.91 yrs,4,"1,710",1 day,0,12.67 yrs,29.68%

Conclusions for decision,End,instant,0,instant,instant,0,119,0.02,instant,1,119,instant,0,instant,2.48%

Freely distributed (by draw) (exclusive movement of the distributor),Registry certificate issued,1 day,0,3.36 days,2.14 wks,0,132,0.03,instant,1,132,2 days,0,1.21 yrs,2.75%

Conclusions for decision,Conclusions for decision,instant,83,2.16 wks,3.65 mths,0,387,0.1,3.39 yrs,6,470,4 days,0,19.47 yrs,8.07%

Publication Certificate issued,Publication Certificate issued,instant,19,1.75 days,5.42 mths,0,271,0.06,3.29 wks,3,290,1 day,0,1.39 yrs,5.65%

Freely distributed (by draw) (exclusive movement of the distributor),DJE,instant,0,2.95 days,2.04 mths,0,666,0.14,instant,1,666,1 day,0,5.37 yrs,13.89%

Start,Petition,instant,0,instant,instant,0,79,0.02,instant,1,79,instant,0,instant,1.65%

Registry certificate issued,Registry certificate issued,instant,8,3.1 days,1.77 mths,0,180,0.04,1.15 mths,2,188,instant,0,1.6 yrs,3.75%

Registry certificate issued,Freely distributed (by draw) (exclusive movement of the distributor),instant,0,instant,instant,0,275,0.06,instant,1,275,instant,0,instant,5.74%

DJE,DJE,instant,269,6.28 hrs,2.27 wks,0,847,0.23,2.07 mths,11,"1,116",instant,0,9.59 mths,17.66%

Petition,Petition,instant,"1,808",6.78 days,11.89 mths,0,"1,537",0.7,36.67 yrs,31,"3,345",1 day,0,62.1 yrs,32.05%

Petition,Publication Certificate issued,instant,402,8.07 hrs,2.71 wks,0,"1,054",0.3,3.97 mths,8,"1,456",instant,0,1.34 yrs,21.98%

DJE,Petition,instant,281,1.77 days,1.02 mths,0,933,0.25,1.43 yrs,5,"1,214",1 day,0,5.89 yrs,19.46%

Publication Certificate issued,Registry certificate issued,instant,65,1.95 wks,1.05 yrs,0,677,0.15,2.9 yrs,4,742,6 days,0,27.71 yrs,14.12%

Petition,Registry certificate issued,instant,51,5.17 days,1.97 mths,0,449,0.1,8.9 mths,3,500,2 days,0,7.07 yrs,9.36%

DJE,End,instant,0,instant,instant,0,52,0.01,instant,1,52,instant,0,instant,1.08%

DJE,Publication Certificate issued,instant,"24,376",1.75 days,1.12 mths,0,"4,690",6.06,115.9 yrs,31,"29,066",1 day,5,139.29 yrs,97.81%

Here are some definition of terms:

Activity: An activity describes tasks or actions, the sequencing of which results in a process. In a process map, an activity can also be called a node.

Transition: A transition is a link from a source activity to a target activity. It indicates that the target activity directly follows the source activity. A transition is also called an arc.

Case frequency of an activity/arc: The case frequency is the number of cases where an activity/arc appears at least once.

Average frequency of an activity/arc: The average frequency is the ratio of the number of times an activity/arc appears in all the cases in the event log to the number of cases in the event log.

Median frequency of an activity/arc: The median frequency is the middle number of cases when the number of cases where the activity/arc appears is arranged from the lowest to the highest.

Max frequency of an activity/arc: The max frequency is the maximum number of times an activity/arc appears in any case.

Min frequency of an activity/arc: The min frequency is the minimum number of times an activity/arc appears in any case.

Total frequency of an activity/arc: The total number of times an activity/arc appears in the event log.

Average duration of an activity/arc: The average time to from the beginning of an activity/arc to the completion of the activity/arc

Median duration of an activity/arc: The median duration of an activity or arc.

Max duration of an activity/arc: The maximum duration of an activity or arc.

Min duration of an activity/arc: The minimum duration of an activity or arc.

Total duration of an activity/arc: The total duration of an activity or arc in the event log.

Average cost of an activity/arc: The average cost when executing an activity or arc.

Median cost of an activity/arc: The median cost when executing an activity or arc.

Max cost of an activity/arc: The maximum cost when executing an activity or arc.

Min cost of an activity/arc: The minimum cost when executing an activity or arc.

Total cost of an activity/arc: The total cost of executing an activity or arc in the event log.

Bottleneck: A bottleneck is the transition with the highest total duration. It indicates that the target activity is unable to begin the activity quickly. It can also be the activity that has the highest total duration. Here it indicates that the activity takes longer to complete.

Note:

The Activity table and Transition table should be studied meticulously and holistically.

The bottleneck can be determined from the Transition table and the Activity table.

- From the Transition table, the bottleneck is the target activity (not the source activity) of an arc that has the highest average duration. This is usually the major source of bottlenecks.

- From the Activity table, the bottleneck is the activity with the highest average duration. This is another type of bottleneck.

Other bottlenecks are the source activity of a transition with relatively higher average duration or activities with relatively higher average duration.

The most frequent activity is the activity in the activity table with the highest total frequency.

To determine the cause of rework, inspect the transitions table. The cause of rework is an activity that leads to a possibility of transitioning back to a previous activity in a given case. For instance, in the transitions A -> B, B -> C, B -> D, D -> A. Activity "B" is a cause for rework because it leads to "D", the rework and "D" leads to "A" which is being executed another time in a case. In this case, A -> B -> D -> A is the rework loop. Note that there could be more activities in a rework loop.

Before answering the question,

- Read the entire table carefully before answering.

- When comparing for most or least, verify from the Activity table/Transition with the absolute highest or least number (even if they may be outliers), not just a high or low value.

- Double check answers before responding.

- The activity names may be long and similar sounding. Take more care in distinguishing between them.

- The tables may have many rows and columns. Do not focus on the first few activities. Instead, read the entire Activity table and Transition table.

All answers for the questions should be from this prompt. The activities table and transition table should be studied meticulously and holistically.

Write your answer in following order

STEPS:

Here are the steps to accomplish the task. Do not add additional text or explanation outside of the STEPS part.

DESCRIPTION:

Here is a description and reasoning behind the above approach.

ANSWER:

Your answer here

PS: Read and understand this. Do not answer any question for now. The questions follow afterwards.

II. Sample prompt with data schema

You are an expert in process mining and a database engineer. I want you to generate the relevant PostgreSQL queries to answer some questions about the performance of a process.

The performance statistics for every activity and transition in an event log is stored in a relational db. The schema of both tables in the db is shown below.

```
cursor.execute('''DROP TABLE IF EXISTS transitions CASCADE''')

cursor.execute('''DROP TABLE IF EXISTS nodes CASCADE''')

conn.commit()

# SQL to create the 'nodes' table with updated types

create_nodes_table_sql = """

CREATE TABLE IF NOT EXISTS nodes (

    Node TEXT PRIMARY KEY,

    Min_duration FLOAT,

    Total_Rework_frequency INT,

    Average_duration FLOAT,

    Max_duration FLOAT,

    Min_frequency INT,

    Case_frequency INT,

    Average_frequency FLOAT,

    Total_Rework_duration FLOAT,

    Max_frequency INT,

    Total_frequency INT,

    Median_duration FLOAT,

    Median_frequency INT,

    Total_duration FLOAT,

    Relative_case_frequency FLOAT   to store percentages as ratios

);
"""

# SQL to create the 'transitions' table with updated types and foreign key constraints

create_transitions_table_sql = """

CREATE TABLE IF NOT EXISTS transitions (

    Source_Node TEXT,

    Target_Node TEXT,

    Min_duration FLOAT,

    Total_Rework_frequency INT,

    Average_duration FLOAT,
```

```
    Max_duration FLOAT,
    Min_frequency INT,
    Case_frequency INT,
    Average_frequency FLOAT,
    Total_Rework_duration FLOAT,
    Max_frequency INT,
    Total_frequency INT,
    Median_duration FLOAT,
    Median_frequency INT,
    Total_duration FLOAT,
    Relative_case_frequency FLOAT,   to store percentages as ratios
    FOREIGN KEY (Source_Node) REFERENCES nodes(Node),
    FOREIGN KEY (Target_Node) REFERENCES nodes(Node)
);
"""
```

Here are some definition of terms:

Activity: An activity describes tasks or actions, the sequencing of which results in a process. In a process map, an activity can also be called a node.

Transition: A transition is a link from a source activity to a target activity. It indicates that the target activity directly follows the source activity. A transition is also called an arc.

Case frequency of an activity/arc: The case frequency is the number of cases where an activity/arc appears at least once.

Average frequency of an activity/arc: The average frequency is the ratio of the number of times an activity/arc appears in all the cases in the event log to the number of cases in the event log.

Median frequency of an activity/arc: The median frequency is the middle number of cases when the number of cases where the activity/arc appears is arranged from the lowest to the highest.

Max frequency of an activity/arc: The max frequency is the maximum number of times an activity/arc appears in any case.

Min frequency of an activity/arc: The min frequency is the minimum number of times an activity/arc appears in any case.

Total frequency of an activity/arc: The total number of times an activity/arc appears in the event log.

Average duration of an activity/arc: The average time to from the beginning of an activity/arc to the completion of the activity/arc

Median duration of an activity/arc: The median duration of an activity or arc.

Max duration of an activity/arc: The maximum duration of an activity or arc.

Min duration of an activity/arc: The minimum duration of an activity or arc.

Total duration of an activity/arc: The total duration of an activity or arc in the event log.

Average cost of an activity/arc: The average cost when executing an activity or arc.

Median cost of an activity/arc: The median cost when executing an activity or arc.

Max cost of an activity/arc: The maximum cost when executing an activity or arc.

Min cost of an activity/arc: The minimum cost when executing an activity or arc.

Total cost of an activity/arc: The total cost of executing an activity or arc in the event log.

Bottleneck: A bottleneck is the transition with the highest total duration. It indicates that the target activity is unable to begin the activity quickly. It can also be the activity that has the highest total duration. Here it indicates that the activity takes longer to complete.

Note:

The Activity table and Transition table should be studied meticulously and holistically.

The bottleneck can be determined from the Transition table and the Activity table.

- From the Transition table, the bottleneck is the target activity (not the source activity) of an arc that has the highest average duration. This is usually the major source of bottlenecks.

- From the Activity table, the bottleneck is the activity with the highest average duration. This is another type of bottleneck.

Other bottlenecks are the source activity of a transition with relatively higher average duration or activities with relatively higher average duration.

The most frequent activity is the activity in the activity table with the highest total frequency.

To determine the cause of rework, inspect the transitions table. The cause of rework is an activity that leads to a possibility of transitioning back to a previous activity in a given case. For instance, in the transitions A -> B, B -> C, B -> D, D -> A. Activity "B" is a cause for rework because it leads to "D", the rework and "D" leads to "A" which is being executed another time in a case. In this case, A -> B -> D -> A is the rework loop. Note that there could be more activities in a rework loop.

Before answering the question,

- Read the entire table carefully before answering.

- Understand the question and generate an accurate PostgreSQL query to get the result.

- Please take note of the column names in both tables.

All answers for the questions should be from this prompt. The schema of the activities table and transition table should be studied meticulously and the relationship between them should be mastered.

Write your answer in following order

STEPS:

Here are the steps to accomplish the task. Do not add additional text or explanation outside of the STEPS part.

DESCRIPTION:

Here is a description and reasoning behind the above approach.

ANSWER:

Your SQL query here

PS: Read and understand this. Do not answer any question for now. The questions follow afterwards

III. Baseline definition of terms

Here are some definition of terms:

Activity: An activity describes tasks or actions, the sequencing of which results in a process. In a process map, an activity can also be called a node.

Transition: A transition is a link from a source activity to a target activity. It indicates that the target activity directly follows the source activity. A transition is also called an arc.

Case frequency of an activity/arc: The case frequency is the number of cases where an activity/arc appears at least once.

Average frequency of an activity/arc: The average frequency is the ratio of the number of time an activity/arc appears in all the cases in the event log to the number of cases in the event log.

Median frequency of an activity/arc: The median frequency is the middle number of case when the number of cases where the activity/arc appear is arranged from the lowest to the highest.

Max frequency of an activity/arc: The max frequency is the maximum number of times an activity/arc appear in any case.

Min frequency of an activity/arc: The min frequency is the minimum number of times an activity/arc appear in any case.

Total frequency of an activity/arc: The total number of times an activity/arc appear in the event log.

Average duration of an activity/arc: The average time to from the beginning of an activity/arc to the completion of the activity/arc

Median duration of an activity/arc: The median duration of an activity or arc.

Max duration of an activity/arc: The maximum duration of an activity or arc.

Min duration of an activity/arc: The minimum duration of an activity or arc.

Total duration of an activity/arc: The total duration of an activity or arc in the event log.

Bottleneck: A bottleneck is the transition with the highest average duration. It indicates that the target activity is unable to begin the activity quickly. It can also be the activity that has the highest average duration. Here it indicates that the activity takes longer to complete an activity.

IV. Upside definition of prompts

Activity: An activity describes a task or action or step that is executed within a process. In a process map, an activity can also be called a node. A specific activity can be observed many times in process executions. Activities have names also called labels.

Case: A sequence of activities defines a process execution. A process execution is also called a case or a process instance.

Log: A log is a collection of data recording cases.

Process map: A process map is a visual representation of a log. A process map is a graph where the nodes represent activities and the arcs represent transitions between two activities.

Activity case frequency: This is the number of times a specific activity is observed in different cases.

Activity total frequency: This is the number of times a specific activity is observed in all the cases.

Activity average frequency: This is calculated as the case frequency of an activity divided the total frequency of the same activity.

Activity median frequency: Considering the number of times we observed a specific activity in a case, this is the median.

Activity max frequency: This is the highest number of times an activity is observed in a single case of a log.

Activity min frequency: This is the least number of times an activity is observed in a single case of a log.

Activity duration: This is the time, in seconds, that an instance of activity takes from start to finish.

Activity average duration: Among all the durations of a specific activity in a log, this is the average duration.

Activity median duration: Among all the durations of a specific activity in a log, this is the median duration.

Activity max duration: Among all the durations of a specific activity in a log, this is the slowest duration.

Activity min duration: Among all the durations of a specific activity in a log, this is the fastest duration.

Activity total duration: Considering all the observed durations of a specific activity in a log, this is their sum.

Activity cost: This is the cost, in euros, of an instance of activity.

Activity average cost: Among all the costs of a specific activity in a log, this is the average cost.

Activity median cost: Among all the costs of a specific activity in a log, this is the median cost.

Activity max cost: Among all the costs of a specific activity in a log, this is the most expensive cost.

Activity min cost: Among all the costs of a specific activity in a log, this is the cheapest cost.

Activity total cost: Considering all the costs of a specific activity in a log, this is their sum.

Activity total rework: This is the number of times we observe a specific activity more than once in a case.

Activity total rework duration: This is the total duration of all the activity instances observed more than once in a case.

Activity total rework duration: This is the total cost of all the activity instances observed more than once in a case.

Activity bottleneck An activity bottleneck is the activity with the highest average duration. It indicates that, in a case, the activity takes a long time to be completed.

Transition: A transition is a directly-follows relation between two activities. In a process map, a transition is represented by an arc having a source node and a target node. A transition can be also called an arc. A specific transition can be observed many times in a log. Transitions do not have names, their name is the concatenation of the source and target node labels.

Source Node: This is where an arc starts.

Target Node: This is where an arc ends.

Transition case frequency: This is the number of times a transition is observed in different cases.

Transition total frequency: This is the number of times a transition is observed in all the cases.

Transition average frequency:This is calculated as the case frequency of a transition divided the total frequency of the same transition.

Transition median frequency: Considering the number of times we observed a transition in a case, this is the median.

Transition max frequency: This is the highest number of times a transition is observed in a single case of a log.

Transition min frequency: This is the least number of times a transition is observed in a single case of a log.

Transition duration: This is the time, in seconds, that a transition takes from start to finish. The duration of a transition is also the waiting time between the source and the target of the transition.

Transition average duration: Among all the durations of a specific transition in a log, this is the average duration.

Transition median duration: Among all the durations of a specific transition in a log, this is the median duration.

Transition max duration: Among all the durations of a specific transition in a log, this is the slowest duration.

Transition min duration: Among all the durations of a specific transition in a log, this is the fastest duration.

Transition total duration: Considering all the durations of a specific transition, this is their sum.

Transition total rework: This is the number of times we observe a specific transition more than once in a case.

Transition total rework duration: This is the total duration of all the transitions observed more than once in a process instance.

Transition bottleneck: A transition bottleneck is the transition with the highest average duration. It indicates that, in a case, the target activity has a long waiting time before being executed. Bottlenecks reduce the overall performance of a process.

V. Questions asked of the prompt.

1. What is the most frequent activity? - Analysis
2. What is the most frequent transition? - Analysis
3. Which activity has the shortest execution time, indicating a potential area of efficiency? - Mine
4. Which transition occurs in the large number of cases? - Analysis
5. Which transition occurs the least in the process? - Analysis
6. What is the average number of activities per case? - Mine
7. What is the average execution time of activity 'Conclusions for decision'? - Analysis
8. What activity is the major bottleneck in the process? - Advanced analysis
9. What are the possible root causes of the bottleneck in my process? – Advanced analysis
10. What other bottlenecks exist in the process? - Advanced analysis
11. Which activity should I automate to get the largest reduction in case duration? - Advanced analysis
12. What activity had the highest average execution time? - Analysis
13. What activity had the highest total duration? - Analysis
14. What are the main causes of rework in this process? - Advanced analysis
15. What changes could I make to reduce the amount of rework? - Advanced analysis
16. What is the average time between activities 'DJE' and 'Publication Certificate issued'? - Analysis
17. What tasks are executed for all cases? - Analysis
18. Is there rework in the process? - Analysis
19. Identify the rework loops in the process? - Advanced analysis
20. How would my process behave if I remove the activity “DJE“? - Advanced analysis

VI. The use of AI

Since this study involved experiments with Large Language Models (LLMs), AI was used to generate the results. As explained in Chapter 5, GPT-4 through the ChatGPT wrapper and Claude Opus 3 through the Anthropic chat interface were used to generate answers to process mining questions. The methodologies and prompts used for the LLMs are outlined in Chapter 4. This is as far as the use of LLMs goes in this study. The process mining questions were informed by relevant process mining literature (see Section 5.2).

VII. Licence

Non-exclusive licence to reproduce the thesis and make the thesis public

I, **David Damola Obembe**,

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis
   Conversational interface for process mining tools using Large Language Models (LLMs),

   supervised by Shefali Sharma and Marlon Dumas.

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

David Damola Obembe

15/05/2024