Explanatory Capabilities of Large Language Models in Prescriptive Process Monitoring

Kateryna Kubrak[1], Lana Botchorishvili[1], Fredrik Milani[1], Alexander Nolte[2,3], and Marlon Dumas[1(✉)]

[1] University of Tartu, Tartu, Estonia
{kateryna.kubrak,lana.botchorishvili,fredrik.milani,marlon.dumas}@ut.ee
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
a.u.nolte@tue.nl
[3] Carnegie Mellon University, Pittsburgh, PA, USA

Abstract. Prescriptive process monitoring (PrPM) systems analyze ongoing business process instances to recommend real-time interventions that optimize performance. The usefulness of these systems hinges on users applying the generated recommendations. Thus, users need to understand the rationale behind these recommendations. One way to build this understanding is to enhance each recommendation with explanations. Existing approaches generate explanations consisting of static text or plots, which users often struggle to understand. Previous work has shown that dialogue systems enhance the effectiveness of explanations in recommender systems. Large Language Models (LLMs) are an emerging technology that facilitates the construction of dialogue systems. In this paper, we investigate the applicability of LLMs for generating explanations in PrPM systems. Following a design science approach, we elicit explainability questions that users may have for PrPM outputs, we design a prompting method on this basis, and we conduct an evaluation with potential users to assess their perception of the explanations and their approach to interact with the system. The results indicate that LLMs can help users of PrPM systems to better understand the origin of the recommendations, and to produce recommendations that have sufficient detail and fulfill their expectations. On the other hand, users find that the explanations do not always address the "why" of a recommendation and do not let them judge if they can trust the recommendation.

Keywords: Prescriptive process monitoring · Explanation · LLM

1 Introduction

Prescriptive Process Monitoring (PrPM) is a family of techniques that recommend runtime interventions to optimize quality, efficiency, or other performance dimensions [20]. These techniques usually rely on Artificial Intelligence (AI)

A. Marrella et al. (Eds.): BPM 2024, LNCS 14940, pp. 403–420, 2024.
https://doi.org/10.1007/978-3-031-70396-6_23

approaches to predict undesirable case outcomes and to evaluate the potential impact of different interventions [2]. For instance, in a loan origination process, PrPM techniques may recommend a manager to perform an additional verification to better estimate the credit risk, or it may recommend making an adjustment to the loan offer to improve the odds of concluding a loan contract. The effectiveness of PrPM systems depends on the extent the recommendations are followed, which itself depends on the workers' ability to understand the rationale behind the recommendations. Previous work has highlighted the challenges of providing understandable and compelling explanations to business users in the context of such systems [28], leading to workers relying on their own judgment and ignoring the recommendations [6].

One way to facilitate the understanding of recommendations produced by AI techniques is to supplement them with explanations. In the field of PrPM systems, and more broadly in other types of recommender systems based on AI techniques, explanations are commonly communicated using plots and numbers (e.g., statistical measures or measures of feature importance). An evaluation of such explanations for predictive process monitoring [30] showed that process analysts struggle with understanding such explanations and that these explanations do not match their information needs [30]. Other studies have shown that dialogue-based systems can enhance the understandability of explanations of outputs of AI systems by allowing users to ask questions from different angles and thus build an iterative understanding of this output [4,21]. Large Language Models (LLMs) are an emerging technology that could facilitate such explanations through dialogue between a system and a user [9]. LLMs can elaborate on plots and numbers, and answer follow-up questions [9].

In this setting, the research objective (RO) of this study is *to design and evaluate an approach for LLM-based explanations of recommendations generated by prescriptive process monitoring techniques*. To pursue this objective, we followed a design science approach [29]. First, we scoped the problem, i.e., enhancing the understandability of explanations of PrPM recommendations. Second, we elicited requirements by drawing up a set of contextualized questions from the eXplainable AI Question Bank [23]. Based on these contextualized questions, we designed and developed our artifact: a prompting method that enables an LLM to elaborate on and explain PrPM recommendations. To evaluate the artifact, we implemented an LLM-based chatbot interface on top of a PrPM tool. Thus, the contribution of this paper is a prompting method to present explanations of recommendations in PrPM, and insights into potential benefits and challenges of designing LLM-based systems for enhancing explainability in PrPM systems.

In the rest of the paper we introduce the concepts used (Sect. 2), outline the method (Sect. 3), describe the artifact's development (Sect. 4), and present our findings (Sect. 5) before discussing them (Sect. 6). Section 7 concludes the paper.

2 Background and Related Work

This section introduces concepts related to explanations in AI systems (Sect. 2.1) and LLMs (Sect. 2.2) and reviews related work on explanations in PrPM systems (Sect. 2.3) and use of LLMs for Business Process Management – BPM (Sect. 2.4).

2.1 Explanations in AI Systems

When constructing explanations of AI outputs for end users, two questions should to be answered; **what** and **how** to explain these outputs [26].

What to Explain. The most widespread way to categorize explanation types is by *How, Why, Why not, What if, How to* and *What else* explanations. These categories were used to develop the explainable AI question bank (XAIQB) [23], a collection of prototypical questions that capture the users' needs in relation to explainability and how to design such explanations. XAIQB has been extended with three more categories: *Output*, *Data*, and *Performance* [24]. In this paper, we use the extended version of XAIQB (Sect. 3.1).

How to Explain. Existing forms of presenting explanation include graphics, images, reports, and texts [4]. The text presentation can be fixed (e.g., plain text) or interactive (e.g., dialogue system) [4,5]. Interactive presentations enable users to ask free-format questions, and thus, are reflective in their expectations of the explanation [21]. Therefore, interactive presentations make explainable user interfaces more accessible [5]. In this paper, we focus on presentation of explanations in an interactive way.

2.2 Large Language Models and Prompt Engineering

Large Language Models (LLMs) refer to transformer models pre-trained on large-scale datasets of text that are capable of performing different natural language processing tasks, e.g., text generation [10]. These models can be fine-tuned to perform specific tasks. Fine-tuning requires a large dataset of labeled and task-specific examples [3]. Such an approach has limitations since the performance of the LLM depends on the quantity and quality of the examples, and requires a copy of the model to be stored for each task [3]. Another approach is prompt engineering. It has gained popularity since the specifics of the task can be defined in a prompt without having to change the LLM itself [3]. Prompt engineering is a process of finding an optimal prompt for a specific task [3]. A prompt is a natural language specification of instructions for the LLM [1]. Commonly, a prompt consists of contextual knowledge, examples, and a task [1,3].

In the field of process mining, several researchers have experimented with prompt engineering [1,3,12,16]. To this end, contextual knowledge has been further specified as process mining knowledge (e.g., how an event log is built), domain knowledge (e.g., definition of a bottleneck in the specific event log), data description (e.g., data structure and specific calculations) [12,16]. We draw from previous research to design the prompt for our study.

Several strategies for prompt engineering have been proposed, such as zero-shot or few-shot settings. In a zero-shot setting, the prompt consists only of a task description, while in the few-shot setting, examples are provided [3]. If the LLM response contains errors or inconsistencies, self-reflection can be used [17]. Self-reflection involves adding a verification layer to the generated response by, e.g., asking the model "Did you pay attention to the instructions?", after which a refined response is generated. Other research also suggests "conversational" strategies to improve the outputs in terms of their style and structure. For instance, giving LLM an identity (e.g., "You are a data scientist.") has shown to improve the quality of LLM's answers [32]. Another example is prompting the LLM to use simple language or ask one question at a time [31].

2.3 Explanations in Prescriptive Process Monitoring Systems

PrPM techniques can be classified into three categories: guiding-, correlation- and causality-based [20]. Guiding techniques produce recommendations based on how similar cases were handled in historical traces. Correlation-based techniques produce recommendations based on predictions of case outcomes [8,18], whereas causality-based methods estimate the effect of an intervention, such as CATE (Conditional Average Treatment Effect) [2]. In this study, we focus on providing explanations for correlation- and causality-based approaches, which typically rely on black-box AI techniques to generate and rank recommendations.

PrPM research has largely focused on optimizing the effect of the recommendations they generate, assuming that the workers will always accept and implement these recommendations [20]. In reality, though, users may or may not act on a recommendation. Thus, providing understandable and compelling explanations to supplement the generated recommendations is an important concern. In [28], the authors develop visualizations to display an expected KPI when following or ignoring a recommendation. Another study utilized SHAP values to explain predictions in ongoing cases at run-time [11]. Another study conducted a user evaluation of explanations in predictive process monitoring [30]. They found that even analysts with foundational knowledge of BPM and ML struggle with understanding and drawing conclusions from metric-driven explanations and visualizations [30]. We extend this field by examining how LLMs can be used to explain recommendations provided by PrPM techniques.

2.4 Large Language Models for Business Process Management

Recent studies have explored the use of LLMs for BPM tasks [7,12,16,19]. Grohs et al. [12] study the use of LLMs to discover process models from textual descriptions and to identify tasks suitable for automation. Klievtsova et al. [19] conduct an analysis of chatbots for conversational process modelling. Jessen et al. [16] focus on prompt engineering for translating process-related questions into SQL-queries that are applied to the database where an event log is stored. In a recent study, Fahland et al. [7] explore LLMs' capabilities to explain business processes. In their study, the authors focus on prompting an LLM with a causal component

so that it can trace back the chain of events in a business process and provide an indication of why something happened. To the best of our knowledge, our study is the first one to focus on the use of LLMs in the context of PrPM.

3 Research Method

To achieve our research objective, we followed a design science approach [29] (Fig. 1 shows an overview). The output of following the DS approach is a prompting method that enables the LLM to elaborate on and explain recommendations. First, we identified the need to enhance explanations of PrPM recommendations to aid understandability with the help of LLMs (described in Sect. 1). Then, we analyzed explainability questions that end users might have for PrPM outputs (Sect. 3.1). This phase resulted in a set of solution requirements. Next, we proceeded with the design and development of the solution (Sect. 3.2) by iteratively developing a prompting method that fulfilled the requirements. We implemented a prototype of an LLM-based chatbot interface on top of an existing PrPM tool to evaluate it. Thus, in the last phase, we conducted an evaluation with process analysts by having them interact with the chat (Sect. 3.3). In particular, we evaluated the generated explanations, end users' perception of the explanations, and their approach to interacting with the chat. We focused on these goals because our aim was to assess the quality of the output and its understandability. Below, we describe the phases in detail.

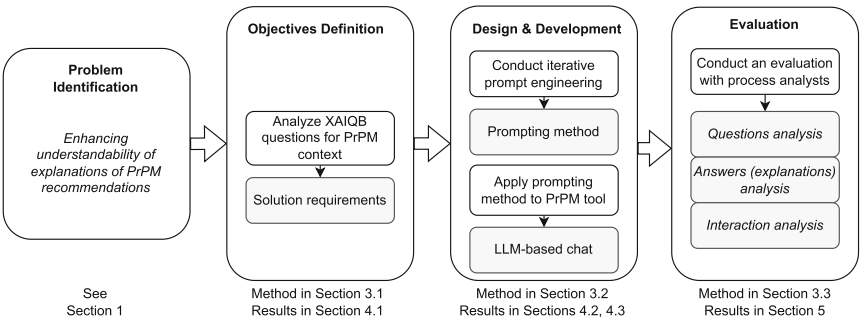

Fig. 1. Research process.

3.1 Objectives Definition

To identify requirements for the design of the solution, we started by analyzing the needs of end users to understand explanations in the context of prescriptive process monitoring. With this aim, we consulted the explainable AI question bank (XAIQB). We identified XAIQB as suitable because it is a collection of prototypical questions that aid in designing question-driven explanations [23,24].

XAIQB consists of ten categories of questions. For instance, the category "Output" includes questions such as "What kind of output does the system give?". We used XAIQB to derive questions for PrPM context. For example, we contextualized the aforementioned question into "What are the recommendations prescribed by the techniques?" Such a question would help with understanding the differences in recommendations prescribed by the prototype.

We also utilized XAIQB's suggestions on ways to design explanations [24]. For instance, to answer questions from the "Output" category, XAIQB suggests to "describe the scope of output or system functions" and, if applicable, "to suggest how the output should be used for downstream tasks or user workflow" [24]. Thus, in the scope of PrPM, a description of recommendation types and their differences could be given. With this in mind, we mapped each category of questions with the possible explanations that could be given as responses. Then, for each question and possible explanations, we formulated a prototypical output that could later be used as examples in the prompt.

3.2 Design and Development

We developed a prompting method based on the requirements elicited in the previous phase. We based the initial prompt on existing literature (see Sect. 2.2). It consisted of context, data description, general conversational rules, task, and examples. The examples were constructed based on the mapping of questions and ways to explain from the previous subsection. Then, we submitted the prompt to ChatGPT and asked whether there was any other information required to fulfill the task. From the response, we edited the prompt, added examples, and iteratively tested and improved it based on the feedback. More detailed description of this process and outcomes are described in Sect. 4.2. To evaluate the prompting method, we implemented a prototype of an LLM-based chatbot interface on top of a PrPM tool (detailed description in Sect. 4.3).

3.3 Evaluation Design

Setting. The participants asked their questions in the developed chat. For the evaluation, we used a synthetic event log of a claim management process. The event log contained 600 cases, 91 case variants, and 4.9K activity instances. The same case was reviewed in all interviews. It was an ongoing case that had a duration of 4 days. The parameters set in the PrPM tool were as follows. The positive outcome of the process was set as a duration under 14 days. In case's current state, the activity "Prepare Claim Settlement" was completed. The PrPM tool produces three different types of recommendations based on previous work (cf. Sect. 2.3). The guiding recommendation was to "Approve Claim Settlement", the causality-based was to "Amend Claim Settlement". A correlation-based recommendation was not prescribed. The focus of the evaluation was on the explanations provided by the chat and not the recommendations themselves.

Participants. The goal of the evaluation was to (1) assess users' perception of generated explanations based on the prompting method, (2) assess users' interaction with the chat. To pursue these goals, we used a mixed-methods approach that consisted of contextual interviews and a survey. Contextual interviews are a method from contextual design that enable studying a use of technology in context [14]. In our case, we observed the individuals while they interacted with the chat (cf. second goal above).

The participants were process analysts, working with process analysis internally at a company or as a consultant. We targeted process analysts as they are skilled in using a variety of process mining techniques [25] and can be expected to benefit from the outputs of PrPM quicker. We selected 12 individuals with a diverse set of experience in process analysis, experience in LLMs, and from different domains. They had, on average, 6.5 years of experience with process analysis and described their experience with LLMs as either slightly (7/12 participants) or fairly well (5/12) experienced. The participants were anonymized as "P-[number]" in the reporting[1]. On average, the interviews lasted for 21 min.

Data Collection. The interviews were conducted by one of the authors. First, participants were introduced to the goals of the study. Then, they received a link to the case. Their task was to review the case and the given recommendations. Then, assuming this was a new system deployed at an insurance company, the participants were instructed to ask questions in the chat about the recommendations. We asked the participants to comment out loud why they were asking a question and whether the answer was satisfying. Participants were also asked to fill out a survey to rate their satisfaction with the explanations. The survey followed the explanation satisfaction scale developed by Hoffman et al. [13].

Analysis. For each interview, we analyzed (1) the spoken interaction between the participant and the interviewee, and (2) the conversation between the participant and the chat, because our aim was to to assess the quality of LLMs output and its understandability. We saved the participant-chat conversation into a separate document and added our observations for each message exchange. Such observations included e.g., the participants' comments on the chat's responses (such as whether they were confusing). These observations helped with the coding of the chat's responses.

The participants' questions and the chat's responses were coded. The coding scheme for the participants' questions was based on the question categories from XAIQB (see Sect. 3.1) (e.g., *Output, Why, How*, etc.). The goal was to investigate what questions the participants asked compared to the questions in XAIQB. For the explanations (chat's responses), we combined deductive and inductive coding. It is common practice in the context of explanations to develop a set of metrics for the explanations that is applicable to the aims of the study [13]. First, we searched for literature on evaluating explanations, particularly, textual explanations. We discovered several systematic review studies on devel-

[1] The detailed overview of study participants is given in Supplementary Material: https://doi.org/10.6084/m9.figshare.25415290.v1.

oping explanation characteristics [13] and their evaluation [27,33]. Then, we went through the explanations to identify codes emerging from the data. Having combined our characteristics with that of the literature, we arrived at the following categories: *"Coherency"* – whether the explanation is internally coherent (how well the parts of it fit together), *"Relevancy to the question"* – whether the explanation answers the question, *"Completeness"* – whether there are gaps in the explanation, *"Correctness"* – whether the data in the explanation is correct, and *"Compactness"* – whether the explanation is repetitive or redundant.

Two authors of the paper conducted the coding independently. They each coded a portion of the dataset, and through multiple rounds arrived at an acceptable agreement score. Cohen's Kappa for questions coding was 0.65 (substantial) and for explanations coding between 0.47–0.5 (moderate). The coding scheme can be found at https://doi.org/10.6084/m9.figshare.25415290.v1. We also analyzed the survey responses. For each question, we calculated the number of responses in each of the Likert-scale categories and visualized it. We treated the result as an additional qualitative data point.

4 Artifact

In this section, we present the artifact: a prompting method to present explanations of recommendations in PrPM. Section 4.1 outlines the elicited requirements, Sect. 4.2 describes the development of the prompting method, and Sect. 4.3 – the integration of the method into the PrPM prototype.

4.1 Elicited Requirements

To elicit requirements, we used XAIQB [23], which categorizes explainability questions into 10 groups. We tailored these questions to the specific context of PrPM and mapped them to potential explanations, serving as examples to be included in the prompt. Table 1 presents an excerpt of this mapping.

Table 1. [Excerpt] Mapping of explainability questions and ways to explain. For full mapping, see: https://doi.org/10.6084/m9.figshare.25415290.v1

Category	Questions	Ways to explain	Prototypical output
Data	What is the size of the event log?	Number of cases in the event log	The event log consists of [number] of cases.
Performance	Why should I believe that the predictions are correct?	Provide performance metrics for the models (accuracy, precision, recall)	The accuracy of recommendations is on average [number].
How	How does the system make predictions?	Describe how the three different algorithms work	The tool provides three different recommendation types: next best activity, alarm and intervention. [...] The intervention is produced using Uplift Modeling package CasualLift to get the CATE and probability of outcome if the intervention is applied or not.

Further categories and questions are described in supplementary material.

Based on the mapping, we elicited the following functional (FR) and non-functional (NR) requirements:

- FR1: The chat's answers should contain correct data from the PrPM outputs.
- FR2: The chat's answers should contain relevant content based on recommendation type.
- NR1: The chat should always provide a response.
- NR2: The chat should respond to the user's question within near-real time.

FR1 refers to the need for the LLM to query correct data from the database to include the relevant data in the prototypical output (e.g., the number of cases). FR2 relates to giving correct information about the techniques that prescribe the different recommendation types (cf. the example of the prototypical output in the category "How" in Table 1). Thus, the LLM would have to correctly match the information.

4.2 Prompting Method

Based on the mapping and requirements introduced in the previous subsection, we designed the prompting method. The initial prompt, drawing from the literature (Sect. 2.2), consisted of context, data description, general conversational rules, task, and examples. Context included specifying the domain (process mining) and details about the PrPM tool, such as which techniques are used, the workflow, and the input parameters. Data description related to the structure of the database connected to the PrPM tool (described in the next subsection). The task for to the LLM was to answer questions about the PrPm tool's recommendations and query the database to obtain the required data for the answers.

To answer the user's questions, the LLM needs to query the database where case data and recommendations are stored (FR1). We conducted three tests to identify how to best represent the query examples in the prompt. For the tests, we used three variations: #1 no examples, #2 example question and steps for making the query, #3 example question, steps for making the query, and the query itself. Prompts #1 and #2 produced incorrect queries by returning the entire case data and an empty response, respectively. For both these prompts, the LLM sometimes queried the files collection instead of the cases collection. Variation #3 produced correct responses. Therefore, we designed the prompting method to include a question, steps, and a query since it proved to work correctly.

Table 2 details the overall structure of the prompt. FR2 did not require querying the database because explanations for different recommendation types are already provided as text within the component "Examples" in the table.

Table 2. [Excerpt] Components of the prompt with text excerpts of each component. For full prompt and full prompt engineering report see: https://doi.org/10.6084/m9.figshare.25415290.v1

Component	Text (excerpt)
Context	[PrPM tool] uses three algorithms to generate prescriptions for business processes... The [PrPM tool] workflow involves: Uploading an event log. Defining column types. Setting parameters... The key parameters are: Case Completion: An activity that marks the end of a case, e.g.,'Application completed'...
Data description	- Description of MongoDB files collection - Description of cases collection
General conversational rules	When answering, use simple language for the explanations. Do not mention the database or show raw data in your responses. ...
Examples	QUESTION: What is the size of the event log? ANSWER: The event log consists of <nr_of_cases> of cases. QUERY: [query example] STEPS: Run the query with function query_db to find the number of cases in this event log.
Task	Your role is to answer questions about [PrPM tool] recommendations and query the database for specific case or event log information.

4.3 Prototype Integration

To evaluate the prompting method, we developed a prototype of an LLM-based chatbot interface on top of a PrPM tool (see Fig. 2). The PrPM tool prescribes recommendations which are stored in the database. For each case, there may be up to three recommendations prescribed (guiding, correlation-based, causality-based; see Sect. 2.3). Users can upload an event log and receive recommendations in running cases. The generated recommendations are displayed together with case attributes in the UI.

When the user asks a question (1), the case ID and the question are sent to the backend (2). It then uses the OpenAI API thread endpoint whenever the user creates a new thread (chat). For each question, the backend creates a new run using OpenAI API endpoint (3). The run is configured to include the thread ID (specific to a case) and assistant ID. The backend also configures the run to overwrite the assistant instructions (the prompt) by appending the event log, case structures, and their respective IDs to the run instructions. If a question requires querying the database, OpenAI provides the backend with the function arguments (4) and the backend queries the database (5). The backend then sends the function output to OpenAI (7), which takes the question and function

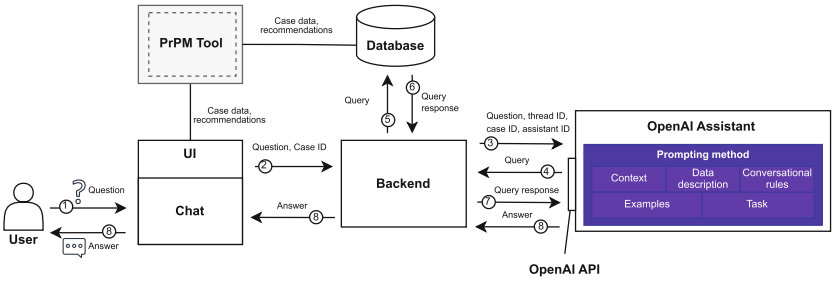

Fig. 2. High-level overview of the prototype: the interaction between the user with the chat interface which displays the answers generated by the LLM based on the prompt.

output and produces the answer (8). For more information on the prototype, see: https://doi.org/10.6084/m9.figshare.25415290.v1

5 Evaluation Results

In this section, we introduce the results of the evaluation (see Sect. 3.3 for evaluation protocol). We divide the results into three parts: participants' questions, chat's answers, and participant-chat interaction. Questions asked and answers given can be viewed at: https://doi.org/10.6084/m9.figshare.25415290.v1

Participants' Questions. The majority of questions were about the *"Output"* (55%). For example, questions aiming at clarifying specific to PrPM terms, such as "CATE score", "positive outcome", and "intervention". Next, several questions related to the recommendation to amend the claim settlement. Specifically, the participants asked for what, more precisely, to amend in the claim settlement. The participants also asked questions about the outcomes of the case, such as what the last activity of the case was predicted to be and when the case is predicted to finish. Last, there were questions relating to the potential impact of the recommendations in terms of specific measures such as hours or cost savings.

The category *"Why"* contained 18% of the questions. The participants wanted to know why a certain recommendation was prescribed. As there were two recommendations given for the case, participants asked questions related to why they should prefer one recommendation over the other.

The third category was *"Others"* with 12% of questions. This category consisted of questions that did not directly relate to the recommendations. For instance, some of the questions were about the contextual information about the case, e.g., what kind of documents were supplied in the claim, or if there was a legal basis for amending the claim settlement. Another set of questions was about case statistics not provided as part of the output, such as the cycle time of the case or how the case compared to others in terms of performance.

The participants asked comparatively few questions related to the categories of *"How"* (7%), *"What if"* (4%), *"Data"* (2%), and *"How to be that"* (1%). We

did not record any questions in categories *"Performance"* (this category refers to the performance of the techniques, e.g., their accuracy), and *"How to still be this"*.

Chat's Answers. The chat provided timely responses (NR2), but failed to provide a response once (NR1). However, this was due to an error in the prototype back-end. Upon refreshing, the chat worked again.

Based on our coding, the distribution of explanation characteristics was as follows: coherency – 98%, relevance to the question – 94%, completeness – 94%, correctness – 75%, compactness – 85%. While coherency, completeness, and relevance of the question were relatively high, the other two characteristics require improvement. Our observations on compactness indicated that in some interviews, most explanations were compact, while in others, most were not compact. One reason for this could be that, in the implementation, every interaction is recorded in a new message thread, i.e., the chat does not have access to responses from earlier interviews. If the chat's first response is long, and the participant does not provide feedback on it, the chat continues to respond similarly.

In regards to correctness, we observed two issues that contributed to incorrect answers. First, sometimes, when receiving a question that would require querying the database, the chat did not do so, but still provided a confident answer. For instance, P-06 asked: *"Can you give me statistics about how the duration of the approval of claim settlement correlates with successful and unsuccessfull (sic.) claims settlement?"*, to which the chat responded: *"The statistics from the information available show that when the approval of a claim settlement takes 4 to 6 days, the claims are successfully settled. These durations correlate with a positive outcome for the claims that adopted this duration window for settlement approval."*. Hence, the response that it takes 4 to 6 days was not derived from the database. Second, sometimes when the chat ran a query on the database, it got certain data but it was incorrectly matched with a question. For instance, P-09 asked what the probability of the next activity after"Prepare Claim Settlement" was. The chat responded: *"The probability of the next activity happening after"Prepare Claim Settlement" is 90.14%, as the suggested next activity is "Approve Claim Settlement""*. However, 90.14% related to the accuracy of the prediction, and not the probability. With these results, FR1 is partially fulfilled and can be improved in the future. For FR2, the chat drew the necessary information from the examples, and its answers were correct.

Interaction. We observed that participants had different approaches to starting the conversation. The majority (8/12) first studied the case and recommendations, and then formulated a question to clarify something they did not understand. One participant opened the chat right away and asked a general question about the case performance before taking a closer look at the case. Two participants started the conversation by asking about issues in the case they should address. Last, one participant reversed the interaction with the chat by asking how it could help them, basing the next question on its response. Related to the last approach, two other participants later suggested adding a feature with pre-defined clickable questions to the chat.

On two occasions, the participants disagreed with the chat's response and wrote it in a follow-up message. In this regard, P-01 wrote, *"I think this is not true"*, and P-04 wrote, *"you misled me :-)"*. In both cases, the chat corrected itself and provided the correct data in the response.

Sometimes, the chat seemed to have issues with correcting the participants when they were wrong. Instead, it gave an answer that was adapted to the participants' incorrect interpretation of the data. For example, in the given case, it was indicated that an alarm was not raised. However, P-01 interpreted it incorrectly and asked the chat: *"Why was there an alarm? The probabaility (sic.) was 0.3522."* (the alarm threshold was set to 0.7). Instead of correcting the participant, the chat responded: *"An alarm was raised because the probability of a negative outcome calculated by the random forest algorithm was 0.3522. Even though this probability does not exceed the specified threshold that generally triggers an intervention, it still represents a significant risk that may warrant closer attention or a review of the case to prevent an undesirable outcome."*

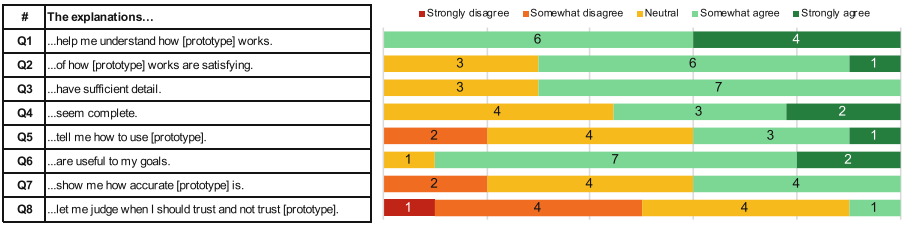

Fig. 3. Survey results.

Overall Explanations Perception. The survey was filled out by 10 out of 12 participants, indicating a response rate of 83%. The participants' perception of the explanations was generally positive, with a few being rather neutral (see Fig. 3). A comparatively low indicator is the participants' trust in the recommendations provided by the prototype. This could be related to the correctness of the explanations which could be improved.

6 Discussion

6.1 Implications for Research

Our findings indicate that participants sought detailed information about activity recommendations. For instance, for the recommendation to amend the claim settlement, the participants wanted to know what exactly to amend. This indicates a need for PrPM techniques to take the parameters of the activity, and not only the activity label, into account. Furthermore, several participants asked about the potential impact of a recommendation on the case outcome in terms

of temporal or monetary value. These observations indicate the importance of incorporating causal aspects into PrPM. More specifically, to provide causal recommendations as in, for instance, [2]. Thus, an avenue for future research is to improve causal recommendations for PrPM.

Several participants also asked questions about why an action was recommended. Such questions could also be addressed by incorporating a causal component. Recent work proposes an approach to generate LLM-based explanations where one input can be a causal view that shows execution dependencies among the activities in a business process [7]. Another example of explanations addressing the "why" is to highlight which attributes of the case are responsible for the given prediction. Here, SHAP values could be used, as exemplified in [11]. Yet another way would be through counterfactuals, which would allow the end users to understand what would need to be done to achieve a desired outcome [15]. Thus, future work could combine LLMs with methods for explainability.

Prior research suggests that process analysts, even those with a foundational understanding of machine learning, can encounter difficulties interpreting metric-driven explanations and visualizations [30]. To address this challenge, this work proposes the use of LLM-based explanations as a potential avenue for future research. An experiment could be designed to compare the understandability of LLM explanations with established methods. These methods might include techniques like SHAP values for visualizing case attributes influencing predictions [11], or tables and graphs that depict expected KPI values under different scenarios (no recommendation, best recommendation, etc.) [28]. This experiment could be further extended to compare the effectiveness of "raw" metric-driven visualizations with those supplemented with LLM-based explanations (such as proposed in the previous paragraph).

The findings showed that the participants were mostly satisfied with the explanations and found them useful. However, the evaluation focused on recommendations for individual cases. One future research direction involves expanding the interface to encompass process-level insights. This could include functionalities for viewing aggregated data (e.g., total active recommendations, number of recommended cases) to provide a broader process perspective for analysts, aligning better with their needs as highlighted by [25]. We also observed that many questions were about *Output* and *Why*. Particularly about why a specific recommendation was given in a case. This information could be equally valuable for operational workers who make case-specific decisions [6]. Therefore, another avenue for future research is to conduct an evaluation of the LLM explanations for specific-case recommendations with operational workers.

Our findings indicate that the correctness of the explanations could be improved. Although the chat corrected itself when being nudged to do so, it is important to secure correct responses. Particularly, the chat should query the database for each question requiring data to be included in the explanation and obtain correct data. When designing the prompt, we ran tests with different ways to represent examples to ensure the querying of the database. We also tested reformulations of the same question (see prompting protocol in the sup-

plementary material in Sect. 4.2). However, we did encounter instances where the LLM generated incorrect information, often referred to as "hallucinations". To address this limitation, one avenue for future research involves incorporating a verification layer, as proposed by [17]. Another potential factor contributing to these "hallucinations" could be the specific LLM chosen for the prototype. Therefore, a further direction for exploration would be to conduct an experiment comparing the performance of different LLM models for the correctness of the responses.

6.2 Implications for Practice

Several participants suggested adding template questions to the chat, with one asking how the chat could help them. This indicates a wish for guidance on what questions the chat can answer, aligning with [22]. Embedding relevant or frequently asked questions in the chat could be helpful.

To better answer questions from the most asked categories, the prompt can be improved to focus less on other categories. This frees up space in the context window of the assistant which, in turn, can be used to provide a more detailed glossary of terms used in the PrPM context. The importance of including a detailed explanation of domain-specific terms in the prompt has also been previously highlighted [16]. Another approach would be to fine-tune the model on examples, and use the prompt solely to describe the context (incl. a detailed glossary), the data, and to provide the general conversational rules.

We also noted that several participants asked questions about case performance (e.g., cycle time, case performance in comparison to others, happy path). Such data helps get contextual information around the recommendations. Therefore, the prompt could be modified to include such data. However, this requires either ensuring that the LLM would be able to calculate e.g., cycle time, or ensure access to case performance data.

6.3 Limitations

In our study, we used a design science approach [29]. There are certain limitations associated with different stages of this approach. First, we used an LLM. Due to its generative nature, LLMs display limited reproducibility. To mitigate this concern, we included participant conversations as supplementary material. Additionally, the study only used one LLM. Future work could explore the impact of applying the same prompt to different LLM models for comparison. The data used for explanation generation was also limited to a single event log and case. However, the primary focus was on evaluating the quality of the explanations themselves, rather than understanding the recommendations within a broader context. The selection of interview subjects may introduce recruitment bias. To reduce this limitation, we selected participants that cover a broad range of experiences and backgrounds. To mitigate a threat of misinterpreting qualitative data due to bias or subjectivity, we involved two coders and calculating the reliability. Finally, the exploratory nature of the study limits the generalizability of findings

beyond the specific context investigated. We accepted this limitation as our aim was to design and evaluate an approach for LLM-based explanations within the PrPM context.

7 Conclusion

In this paper, we presented the design and evaluation of an approach for LLM-based explanations of recommendations generated by a PrPM system. To achieve this objective, we elicited user needs for explainability in PrPM using an eXplainable AI Question Bank. We developed a prompting method consisting of context, data description, general conversational rules, examples, and task. To evaluate the prompting method, we implemented a prototype of an LLM-based chatbot interface on top of a PrPM tool. The implications for research point towards the need for further development of causal recommendations in PrPM, as well as causal explanations. Future research of explanations in PrPM may use the guidance of questions asked in our study to cater the explanations to the end-user needs. Practical implications include adding template questions to the chat, improving the prompt specifically for most-asked questions, and enabling questions about case performance-related information.

In future work, we aim to refine the prompting method to nudge the LLM to provide a rationale for each recommendation (i.e. bringing out the "why"). We also plan to conduct a user study to compare the explanations generated by the proposed LLM prompting method versus existing methods from the field of explainable AI, with the aim of eliciting potential synergies between these approaches. Additionally, we aim to evaluate explanations for case-level recommendations with operational workers, to complement the existing evaluation with tactical managers.

Acknowledgements. This research is supported by the Estonian Research Council (PRG1226) and the European Research Council (PIX Project).

References

1. Bellan, P., Dragoni, M., Ghidini, C.: Extracting business process entities and relations from text using pre-trained language models and in-context learning. In: Almeida, J.P.A., Karastoyanova, D., Guizzardi, G., Montali, M., Maggi, F.M., Fonseca, C.M. (eds.) EDOC 2022. LNCS, vol. 13585, pp. 182–199. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17604-3_11
2. Bozorgi, Z.D., Teinemaa, I., Dumas, M., Rosa, M.L., Polyvyanyy, A.: Prescriptive process monitoring based on causal effect estimation. Inf. Syst. **116**, 102198 (2023)
3. Busch, K., Rochlitzer, A., Sola, D., Leopold, H.: Just tell me: prompt engineering in business process management. In: van der Aa, H., Bork, D., Proper, H.A., Schmidt, R. (eds.) BPMDS EMMSAD 2023 2023. LNBIP, vol. 479, pp. 3–11. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-34241-7_1
4. Cambria, E., Malandri, L., Mercorio, F., Mezzanzanica, M., Nobani, N.: A survey on XAI and natural language explanations. Inf. Process. Manag. **60**(1), 103111 (2023)

5. Chromik, M., Butz, A.: Human-XAI interaction: a review and design principles for explanation user interfaces. In: Ardito, C., et al. (eds.) INTERACT 2021. LNCS, vol. 12933, pp. 619–640. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85616-8_36
6. Dees, M., de Leoni, M., van der Aalst, W.M.P., Reijers, H.A.: What if process predictions are not followed by good recommendations? In: BPM (Industry Forum). CEUR Workshop Proceedings, vol. 2428, pp. 61–72. CEUR-WS.org (2019)
7. Fahland, D., Fournier, F., Limonad, L., Skarbovsky, I., Swevels, A.J.E.: How well can large language models explain business processes? CoRR **abs/2401.12846** (2024)
8. Fahrenkrog-Petersen, S.A., et al.: Fire now, fire later: alarm-based systems for prescriptive process monitoring. Knowl. Inf. Syst. **64**(2), 559–587 (2022)
9. Feldhus, N., Ravichandran, A.M., Möller, S.: Mediators: conversational agents explaining NLP model behavior. CoRR **abs/2206.06029** (2022)
10. Feuerriegel, S., Hartmann, J., Janiesch, C., Zschech, P.: Generative AI. Bus. Inf. Syst. Eng. (2023)
11. Galanti, R., et al.: An explainable decision support system for predictive process analytics. Eng. Appl. Artif. Intell. **120**, 105904 (2023)
12. Grohs, M., Abb, L., Elsayed, N., Rehse, J.: Large language models can accomplish business process management tasks. In: De Weerdt, J., Pufahl, L. (eds.) BPM 2023. LNBIP, vol. 492, pp. 453–465. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-50974-2_34
13. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Measures for explainable AI: explanation goodness, user satisfaction, mental models, curiosity, trust, and human-ai performance. Front. Comput. Sci. **5** (2023)
14. Holtzblatt, K.: Contextual design. In: The Human-computer Interaction Handbook, pp. 975–990. CRC press (2007)
15. Hsieh, C., Moreira, C., Ouyang, C.: Dice4el: interpreting process predictions using a milestone-aware counterfactual approach. In: ICPM, pp. 88–95. IEEE (2021)
16. Jessen, U., Sroka, M., Fahland, D.: Chit-chat or deep talk: prompt engineering for process mining. CoRR **abs/2307.09909** (2023)
17. Ji, Z., Yu, T., Xu, Y., Lee, N., Ishii, E., Fung, P.: Towards mitigating LLM hallucination via self reflection. In: Bouamor, H., Pino, J., Bali, K. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 1827–1843. ACL, Singapore (2023)
18. Khan, A., et al.: DeepProcess: supporting business process execution using a MANN-based recommender system. In: Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H. (eds.) ICSOC 2021. LNCS, vol. 13121, pp. 19–33. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91431-8_2
19. Klievtsova, N., Benzin, J., Kampik, T., Mangler, J., Rinderle-Ma, S.: Conversational process modelling: state of the art, applications, and implications in practice. In: Di Francescomarino, C., Burattin, A., Janiesch, C., Sadiq, S. (eds.) BPM 2023. LNBIP, vol. 490, pp. 319–336. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-41623-1_19
20. Kubrak, K., Milani, F., Nolte, A., Dumas, M.: Prescriptive process monitoring: quo vadis? PeerJ Comput. Sci. **8**, e1097 (2022)
21. Laato, S., Tiainen, M., Islam, A.K.M.N., Mäntymäki, M.: How to explain AI systems to end users: a systematic literature review and research agenda. Internet Res. **32**(7), 1–31 (2022)

22. Lee, Y., Kim, T.S., Kim, S., Yun, Y., Kim, J.: DAPIE: interactive step-by-step explanatory dialogues to answer children's why and how questions. In: CHI, pp. 450:1–450:22. ACM (2023)
23. Liao, Q.V., Gruen, D.M., Miller, S.: Questioning the AI: informing design practices for explainable AI user experiences. In: CHI, pp. 1–15. ACM (2020)
24. Liao, Q.V., Varshney, K.R.: Human-centered explainable AI (XAI): from algorithms to user experiences. CoRR **abs/2110.10790** (2021)
25. Milani, F., Lashkevich, K., Maggi, F.M., Francescomarino, C.D.: Process mining: a guide for practitioners. In: Guizzardi, R., Ralyté, J., Franch, X. (eds.) RCIS 2022. LNBIP, vol. 446, pp. 265–282. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_16
26. Mohseni, S., Zarei, N., Ragan, E.D.: A multidisciplinary survey and framework for design and evaluation of explainable AI systems. ACM Trans. Interact. Intell. Syst. **11**(3–4), 24:1–24:45 (2021)
27. Nauta, M., et al.: From anecdotal evidence to quantitative evaluation methods: a systematic review on evaluating explainable AI. ACM Comp. Surv. **55**(13s), 295:1–295:42 (2023)
28. Padella, A., de Leoni, M., Dogan, O., Galanti, R.: Explainable process prescriptive analytics. In: ICPM, pp. 16–23. IEEE (2022)
29. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manage. Inf. Syst. **24**(3), 45–77 (2008)
30. Rizzi, W., et al.: Explainable predictive process monitoring: a user evaluation. CoRR **abs/2202.07760** (2022)
31. Seo, W., Yang, C., Kim, Y.: Chacha: leveraging large language models to prompt children to share their emotions about personal events. CoRR **abs/2309.12244** (2023)
32. Wei, J., Kim, S., Jung, H., Kim, Y.: Leveraging large language models to power chatbots for collecting user self-reported data. CoRR **abs/2301.05843** (2023)
33. Zemla, J.C., Sloman, S., Bechlivanidis, C., Lagnado, D.A.: Evaluating everyday explanations. Psychon. Bull. Rev. **24**, 1488–1500 (2017)