Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-024-00898-1

RESEARCH

Conversing with business process-aware large language models: the BPLLM framework

Mario Luca Bernardi[1] · **Angelo Casciani**[2] · **Marta Cimitile**[3] · **Andrea Marrella**[2]

Received: 18 March 2024 / Revised: 27 September 2024 / Accepted: 29 September 2024
© The Author(s) 2024

Abstract
Traditionally, process-aware Decision Support Systems (DSSs) have been enhanced with AI functionalities to facilitate quick and informed decision-making. In this context, AI-Augmented Business Process Management Systems have emerged as innovative human-centric information systems, blending flexibility, autonomy, and conversational capability. Large Language Models (LLMs) have significantly boosted such systems, showcasing remarkable natural language processing capabilities across various tasks. Despite the potential of LLMs to support human decisions in business contexts, empirical validations of their effectiveness for process-aware decision support are scarce in the literature. In this paper, we propose the Business Process Large Language Model (BPLLM) framework, a novel approach for enacting actionable conversations with human workers. BPLLM couples Retrieval-Augmented Generation with fine-tuning, to enrich process-specific knowledge. Additionally, a process-aware chunking approach is incorporated to enhance the BPLLM pipeline. We evaluated the approach in various experimental scenarios to assess its ability to generate accurate and contextually relevant answers to users' questions. The empirical study shows the promising performance of the framework in identifying the presence of particular activities and sequence flows within the considered process model, offering insights into its potential for enhancing process-aware DSSs.

Keywords Business process · Decision support systems · LLM · RAG

✉ Angelo Casciani
casciani@diag.uniroma1.it

Mario Luca Bernardi
bernardi@unisannio.it

Marta Cimitile
marta.cimitile@unitelasapienza.it

Andrea Marrella
marrella@diag.uniroma1.it

[1] Department of Engineering, University of Sannio, Piazza Roma 21, Benevento 82100, Italy

[2] Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Rome 00185, Italy

[3] Department of Law and Digital Society, UnitelmaSapienza, Piazza Sassari, Rome 00185, Italy

Published online: 04 October 2024

1 Introduction

AI-Augmented Business Process Management Systems (ABPMSs) represent new human-centered information systems characterized by great flexibility, autonomy, and huge conversational and self-improving capabilities (Dumas et al., 2023). In this context, traditional *process-aware Decision Support Systems* (DSS) are empowered with AI to ensure quick and quality decisions by understanding and explaining the factors behind their choices (Agarwal et al., 2022).

A huge impulse through these kinds of systems can be given by the adoption of *Large Language Models* (LLMs) (Casciani et al., 2024). LLMs are emerging machine learning (ML) models showing a great capability to achieve a variety of natural language processing (NLP) tasks (Ozkaya, 2023). Given their numerous benefits, these models are increasingly employed in several contexts (Ray, 2023), promising great benefits for industries and business functions and revolutionizing the human interaction with management systems (Fahland et al., 2024). Indeed, LLMs are driving organizations towards the paradigm of autonomous enterprises, characterized by the automation of numerous activities and operations. In this paradigm, ABPMSs play a pivotal role in supporting human tasks and decisions throughout the system life cycle. Fahland et al. (2024). Starting from business processes (BPs), LLMs are expected to overrun the local reasoning context, handle different scenarios, and improve the comprehension of business activities within their outcomes (Fahland et al., 2024).

In front of the recognized potential of LLMs to assist human decisions in business contexts (Dumas et al., 2023), this topic is few explored in literature (Fahland et al., 2024), and to the best of our knowledge there is not an empirical validation of the effectiveness of LLMs for process-aware decision support. Starting from these considerations, this study introduces a novel approach to BP analysis and description based on the adoption of LLMs to implement a conversational process-aware DSS. To enhance the conversational abilities of an LLM for answering BP-related tasks, we propose to adopt a process-aware *Retrieval-Augmented Generation* (RAG) framework in conjunction with fine-tuning. This strategy can leverage fine-tuning to extend process- and domain-specific structural and behavioral knowledge leaving RAG in charge of incorporating the contextual knowledge related to the specific user requests. The overall system, called *Business Process LLM* (BPLLM) and fine-tuned on a specific process model, supports the user in a wide range of process comprehension and execution tasks using natural language. In the proposed framework, a LLaMA 2 (Touvron et al., 2023a) LLM is combined with a variant of RAG (Lewis et al., 2020) tailored to deal with the specific aspects of the structural and behavioral representation of BPs. Moreover, a process-aware chunking approach (Object Management Group, 2011) along with a suitable prompting strategy is included. To implement the BPLLM pipeline, we tested different embedding models to investigate the most suitable ones. In addition to the BPLLM pipeline, we propose to fine-tune the LLM to improve its knowledge of the process.

This work investigates the following research questions aimed to evaluate the BPLLM performance within its components and its different settings:

RQ1: *How does the adoption of Retrieval-Augmented Generation (RAG), in the frame of BPLLM architecture, impacts the end-to-end performance?*

RQ2: *What's the influence of chunking and prompting on the performance of the BPLLM?*

RQ3: *How does the choice of process representation format and related embedding model affect the performance of the BPLLM?*

RQ4: *What is the effect of fine-tuning along with the use of RAG on the performance of the BPLLM? Does the number of processes included in the knowledge base of a single LLM impact performance?*

We assessed the capability of BPLLM to generate accurate and contextually relevant answers to users' questions on a food delivery process model. The best BPLLM configuration is also used on different processes to evaluate how the BPLLM performance changes when more process models are used.

In this work, we opted to use *open-source* LLMs for several reasons. Open LLMs offer reduced access costs, guarantee control and stability, and enhance transparency. Researchers can easily inspect and fine-tune these models to their specific domains, fostering collaboration and innovation.

The paper is structured into seven sections. Section 2, reports a description of the related work. Section 3 reports background information useful to better understand and contextualize the proposed approach introduced in Section 4. Section 5 describes the set of experiments carried out and discusses the results. Section 6 details the threats to the validity of the study and Section 7 discusses the conclusions.

2 Related work

Great interest has been pointed in recent years to DSSs given their utility to assist humans in making high-quality decisions based on domain-specific information (Mozannar & Sontag, 2020). Starting from the work proposed in (Schonenberg et al., 2008), some studies focused on the development of recommendation services to drive decisions during process execution by giving suggestions on possible next steps. For example, in (Conforti et al., 2015), the authors introduce a recommendation system able to suggest risk-informed decisions. Some additional studies (Bennett & Hauser, 2013) propose a Markov Decision Process (MDP) (Voorberg et al., 2019) to identify the best decision based on the probability distributions of all possible decisions. The MDP is also used in (Voorberg et al., 2019) to realize a DSS for declarative artifact-centric BP models. Other works focused on the identification of DSSs (Agarwal et al., 2022) that radically evolve to capture the new opportunities derived by the emerging Artificial Intelligence (AI) models (Ali et al., 2023). In this direction, authors in (Agarwal et al., 2022) introduce an end-to-end process-aware DSS able to predict the decisions and explain which factors influence them.

Differently from the described literature, in this study, we focused on generative AI and conversational systems applied to process-aware support systems. The main idea is that the emerging LLM technologies can improve process-based DSSs. According to this, a recent work discussed the possible implications of conversational systems applied to decision support in Process Management (BPM) and Process Mining (PM) (Chapela-Campa & Dumas, 2023). While there are several challenges in training ML models on process data (Ceravolo et al., 2024), the effectiveness of LLMs in performing BPM tasks was thoroughly reviewed (Estrada-Torres et al., 2024) and demonstrated by several works (Grohs et al., 2024; Pasquadibisceglie et al., 2024). Additionally, LLMs proved useful in analyzing and redesigning BPs, yielding more relevant and actionable recommendations for reducing waiting times (Lashkevich et al., 2024), and in transforming business questions and hypotheses written in natural language into executable specifications to generate relevant reports for stakeholders (Berti et al., 2023). Conversational systems can also contribute to presenting the results of PM analysis using a simple natural language by creating a process description from an event log,

or a concise report describing the process performance (Fontenla-Seco et al., 2020; van der Aa et al., 2018; López et al., 2019).

In front of this strong awareness about the importance of exploring the new opportunities of LLMs in the BPM community, the implementation of an LLM-based approach in process-based DSSs is still missed. The present study represents a novelty with respect to the more recent literature since it introduces and experiments with a LLM to improve the efficiency of DSS. The study also proposes, for the first time in the BP representation, the adoption of RAG integrated into the LLM to ensure greater adherence to the specific aspects of the BP domain. Finally, the impact of the proposed approach is evaluated and discussed.

3 Background

3.1 Business process models

A *process model* represents the BP structure, encompassing the activities to be executed and the constraints governing their sequence. It also encapsulates criteria indicating the start and termination of the process, along with details concerning participants, IT systems, and data (Dumas et al., 2013). Several process models have been proposed in recent years (Vaisman, 2013) within their application in real domains (Bernardi et al., 2014; Agostinelli et al., 2022). Among the simplest notations for BPs lies the *Directly-Follows Graph* (DFG) (van der Aalst, 2019). This graphical representation comprises nodes denoting individual activities, while directed arcs delineate the sequential relationships between them, signifying direct succession.

Another common modeling language for BPs is *Business Processing Modeling Notation* (BPMN). It is a standard language, proposed by the Object Management Group (OMG), to design BPs (Object Management Group, 2011). BPMN defines a process model that includes a set of graphical constructs divided into: (i) flow objects, (ii) data, (iii) connecting objects, and (iv) swimlanes. *Flow objects* define the behavior of a process, as the one reported in Fig. 1. They can be classified into *events*, *activities*, and *gateways*. *Events* model the occurrence of states in the real world that are relevant for processes and, more generally, anything that can happen instantaneously (e.g., an invoice has been received). *Activities* represent units of work performed during processes that, differently from the events, have a certain duration (e.g., pay an invoice). *Gateways* are used to represent the split and join behavior of the control flow when there is a need to model specific conditions like mutual exclusion or concurrence.

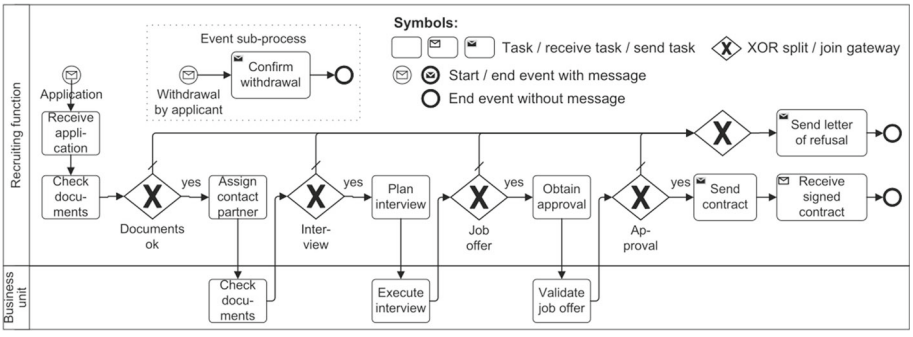

Fig. 1 A process model by (Lohrmann & Reichert, 2016) specified in BPMN

3.2 Large language models and RAG

LLMs represent a significant advancement in NLP, leveraging deep neural network architectures to analyze and generate human-like text. These models undergo extensive training on massive amounts of text data to learn patterns and entity relationships within natural language. Intending to predict and generate coherent text, LLMs estimate the likelihood of tokens or sequences of tokens occurring within larger sequences (Zhao et al., 2023). Over time, LLMs have evolved significantly, transitioning from earlier statistical methodologies to more sophisticated transformer-based architectures like BERT, GPT, and their successors (Vaswani et al., 2017). These models are capable of generating entire documents and have fostered research contributions, including architectural enhancements, training methodologies, context expansion, fine-tuning techniques, and the development of multi-modal LLMs (Naveed et al., 2023). With their escalating size, often containing billions of parameters, LLMs have witnessed substantial performance enhancements, enabling them to excel across diverse NLP tasks. Nonetheless, challenges persist in contextual appropriateness and text accuracy of the output generated by LLMs (McKenzie et al., 2023). These limitations can be attributed, in part, to outdated and biased knowledge embedded within their training datasets (Gao et al., 2024).

Fine-tuning has emerged as a pivotal technique for customizing pre-trained models to suit specific tasks. Indeed, while pre-trained LLMs demonstrate strong performance across a wide range of activities, they may not always excel in specific domains with specialized requirements (Xu & Wang, 2023). Fine-tuning addresses this limitation by updating the parameters of a pre-trained model using task-specific data, thereby adapting the model's knowledge and capabilities to address the nuances of a particular task better. The fine-tuning process typically starts by selecting the pre-trained LLM that better aligns with the task requirements. Next, a task-specific dataset with labeled examples or annotated text data is acquired, to serve as the training data. During fine-tuning, backpropagation iteratively updates the pre-trained parameters based on the dataset to minimize a defined loss function and optimize the model's performance on the target task (Hosseini et al., 2023; Xu et al., 2023a). While fine-tuning offers numerous benefits, it also comes with drawbacks, primarily due to its resource-intensive nature, necessitating substantial computational resources and data. However, these limitations can be mitigated through specific techniques such as *Parameter-Efficient Fine-Tuning* (PEFT) (Xu et al., 2023). PEFT minimizes the number of trainable parameters while retaining comparable performance to full fine-tuning, updating only a limited number of additional parameters or a subset of pre-trained ones.

An encouraging solution to improve LLM accuracy and credibility (especially in knowledge-intensive tasks) and avoid the effort required by complex fine-tuning is represented by RAG (Lewis et al., 2020), which is becoming a popular paradigm in LLM's systems. The underlying idea of RAG is the merging of LLMs' knowledge with specialized, vast, and dynamic data coming from external repositories (Lewis et al., 2020). The initial query prompt is complemented through the external retrieval of pertinent information via search algorithms, which provides further context information (Gao et al., 2024). According to this, RAG combines information retrieval mechanisms with In-Context Learning (ICL) (Dong et al., 2024) to improve the LLM's performance. The approach includes a retriever and a generator (Lewis et al., 2020) and consists of three steps (retrieve, augment, and generate). In the retrieval step, the user query x is used to retrieve relevant context (text documents z) from an external knowledge source by the retriever $p_\eta(z|x)$ with parameters η returning distributions over text documents given x. Using an embedding model, the query is embedded into a vector space and included as the additional context in the vector database. According to

the similarities between vectors and query, the k closest documents from the vector database are selected. In the augment step, the initial query and the obtained additional context are combined into a prompt template. In the last step, the retrieval-augmented prompt is fed to the LLM which generates an answer to the question on the base of the contextual information obtained by retrieved chunks.

3.2.1 LLaMA

LLaMA is a set of open-source LLMs developed by Meta AI, spacing from 7B to 65B parameters (Touvron et al., 2023a). They were trained on publicly available datasets and demonstrated to be competitive with Chinchilla and PaLM-540B. The training was performed on large transformers using a standard optimizer and the training stability was improved by normalizing the input of each transformer sub-layer.

Their evolution is embodied by LLaMA 2, introducing a new breed of pre-trained and fine-tuned models with scales spanning from 7B to 70B parameters (Touvron et al., 2023b). These models have demonstrated improved performance, integrating techniques such as grouped-query attention and Reinforcement Learning with Human Feedback (RLHF) to enhance their performance and safety. The training process for LLaMA 2 adopted a multi-stage methodology, starting with the pre-training of the model using an extensive array of publicly accessible datasets. Subsequently, the model underwent supervised fine-tuning tailored to accommodate dialogue-oriented applications, thereby refining interactions with users.

Another enhancement for open-source models was the release of LLaMA 3 in 2024 (Dubey et al., 2024). These models (8B and 70B parameters) were pre-trained on publicly available datasets and showed to beat (with its 70B version) Gemini Pro 1.5 and Claude 3 Sonnet on most benchmarks, rivaling GPT-4 on several tasks. Subsequently, LLaMA 3.1 was introduced in three variants (8B, 70B, and 405B parameters), significantly expanding the context window up to 128k tokens (Dubey et al., 2024).

3.2.2 Mistral

Mistral constitutes another major family of open-source models. The base version is a 7B-parameter LLM released in 2023, embedding grouped-query attention for cheaper and faster inference. It outperformed LLaMA 2 13B on all tested benchmarks (Jiang et al., 2023). Several improved versions of this model were developed, along with their corresponding "instruct" variants, fine-tuned to follow instructions.

Mixtral 8x7B represented a step forward in the open-source LLMs landscape, relying on a sparse mixture of experts architecture and outperforming both LLaMA 70B and GPT-3.5 in most benchmarks (Jiang et al., 2024). The model embeds 8 groups of "experts" totaling 46.7B parameters, but each token uses only 12.9B parameters, optimizing for speed and cost similar to a 12.9B parameter model.

4 The business process LLM (BPLLM)

In this section, we present our BPLLM framework. Figure 2 shows the operational steps utilized by the BPLLM pipeline for answering queries about BPs. The overarching workflow unfolds in three major phases as follows:

- **RAG Knowledge Augmentation**: Initially, the BPLLM pipeline ingests the target process model, which undergoes the initial stage of chunking. Here, the process model

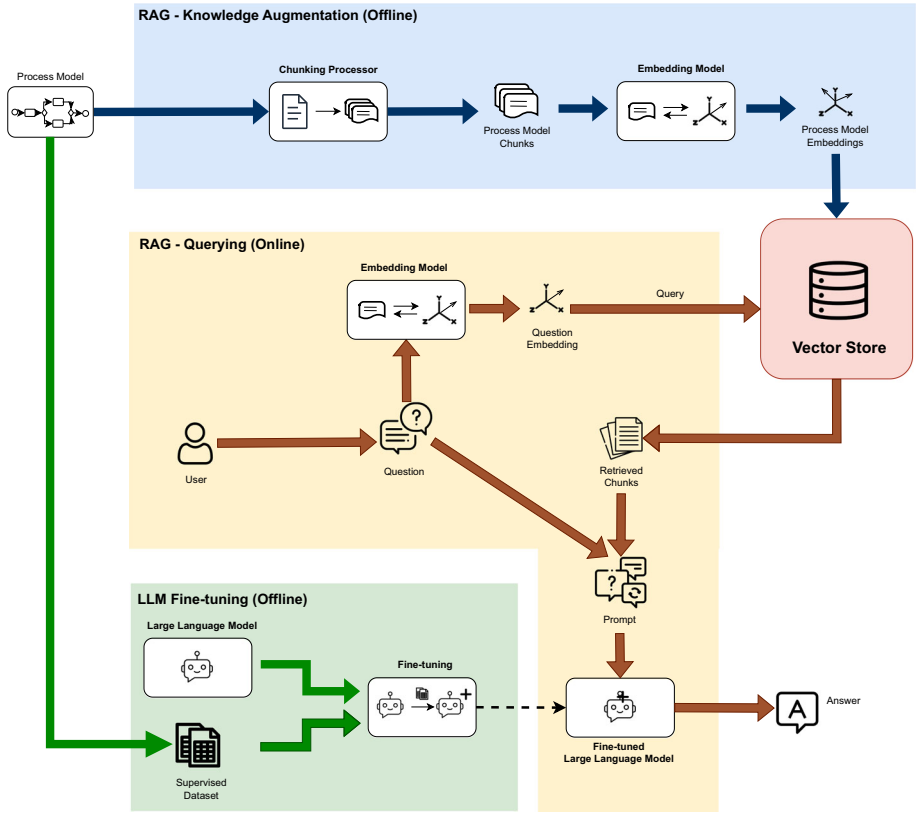

Fig. 2 The overall architecture of the BPLLM framework

representation is segmented into semantically meaningful chunks. Subsequently, the embedding phase processes the previously generated chunks and converts them into embeddings for indexing in the vector database. Following this, the produced embeddings are stored in the designated vector database, to implement subsequent semantic searches based on cosine similarity.

- **LLM Fine-tuning**: The chosen LLM undergoes a process of fine-tuning using a supervised dataset enriched with BP-specific information. This fine-tuning procedure enhances the LLM's understanding of the process and tailors it to the task of responding to inquiries pertinent to the BP model.
- **RAG Querying**: At runtime, during the retrieving stage, a semantic search is conducted to retrieve contextually relevant chunks of the process model from the vector database, based on the user query. These retrieved chunks are then combined with the input query to generate the prompt, which is sent to the LLM to generate the output. Finally, the crafted prompt, derived from the previous stage, is fed into the fine-tuned LLM, which generates an answer containing grounded information concerning the process model.

In the following, we provide an in-depth discussion of each phase within the overall process. From a technological standpoint, the backbone of the pipeline was realized by leveraging the Langchain framework[1].

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

4.1 RAG knowledge augmentation

The BPLLM Knowledge Augmentation is composed of the three stages that are described in this section: (*i*) *Chunking*, (*ii*) *Embedding*, and (*iii*) *Storing*.

4.1.1 Chunking

The initial step in the proposed pipeline takes the BP model in input, aiming to generate multiple relevant chunks for easier comprehension by the LLM in subsequent responses. We considered both a natural language DFG representation and a BPMN representation of the process model.

Due to the limited context windows of LLMs, i.e., the number of tokens the LLM can receive in input to generate the output text, we are prevented from passing large process models in the query. For this reason, *chunking* is an essential operation for RAG and it consists of breaking down extensive contents into smaller, manageable segments. Efficient chunking optimizes the relevance of responses by ensuring that only pertinent context is provided to the LLM. Balancing performance and accuracy is crucial in determining the optimal chunking strategy, as excessive chunking can disrupt content flow. Striving for meaningful chunks and trying to avoid unnatural segmentation of the process model, we evaluated different chunking strategies for each considered process model format. Referring to the DFG textual process description, we considered both fixed-size and recursive chunking. Recursive chunking divides text hierarchically using separators until desired chunk sizes or structures are attained. We evaluated also fixed-size and recursive chunking for the BPMN process.

BPMN-specific chunking For the BPMN representation, we designed an additional chunking technique to intelligently segment the model without sacrificing the semantics of the process model. Our chunking processor parses BPMN files by splitting them in correspondence with relevant tags while disregarding irrelevant graphical elements to preserve contextual relevance. Notably, it scans the BPMN file to isolate self-contained semantic chunks defined by the content within BPMN tags, adding them to the final list to store in the vector database. Figure 3 illustrates its execution on a BPMN file. When processing the depicted BPMN extract, the chunking processor identifies the start of the `<task>` tag and generates a corresponding semantically meaningful chunk. Then, it continues its analysis by isolating chunks related to BPMN elements such as the Exclusive Gateway and the Send Task.

4.1.2 Embedding

The *embedding* phase plays a crucial role in the representation of the input data: it transforms the process model chunks generated in the chunking step into process model embeddings, following a numerical format that can be efficiently processed and stored in a vector database. In particular, the embedding phase transforms the raw input into dense, low-dimensional vectors, capturing semantic information and contextual relationships. These embeddings serve as the foundation for downstream tasks, such as retrieval and generation, in the pipeline. The choice of embedding model significantly impacts the effectiveness of the BPLLM pipeline. Each embedding model has its strengths and weaknesses, depending on the nature of the input data and the requirements of the downstream tasks.

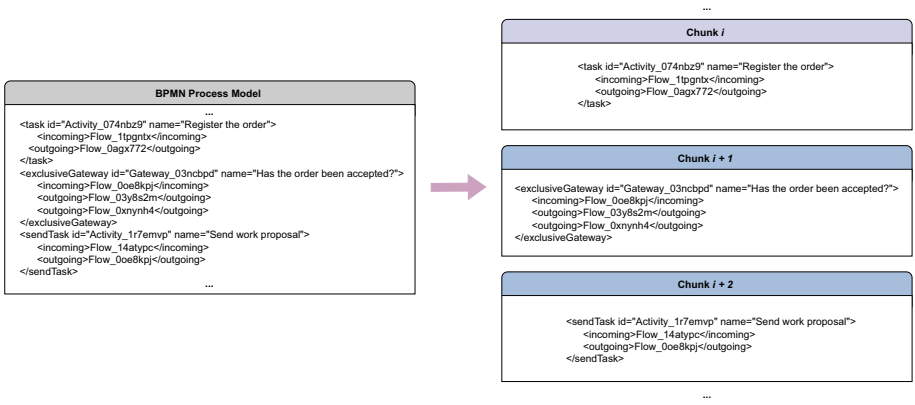

Fig. 3 An example of application of the BPMN-specific chunking technique

We used various embedding models suitable for semantic representation obtaining the best results with the *all-MiniLM-L6-v2*[2] model.

4.1.3 Storing

In the storing phase, the process model embeddings are stored in a dedicated *vector index*. This vector index plays a crucial role in supporting similarity search operations and, consequently, in facilitating efficient retrieval of process model chunks.

In this study, the adoption of different vector index solutions has been explored: after careful consideration, we opted for qDrant[3], an open-source vector database designed for storing and querying high-dimensional embeddings. This decision aligned with our broader objective of leveraging *open-source* technologies throughout the BPLLM implementation. Starting from qDrant, the vector index leveraging *cosine similarity* is used for semantic search. Cosine similarity is vastly used and well-suited for measuring the similarity between high-dimensional vectors, making it ideal for our use case in process-related information retrieval. Finally, the local connection to the vector index is obtained using gRPC[4] as the communication protocol for efficient and reliable communication between our pipeline and the qDrant database.

4.2 LLM fine-tuning

To generate the answers, we employed several language models from the Llama and Mistral families, granted for use by Meta AI, Mistral AI, and HuggingFace. These open-source LLMs are well-suited for natural language understanding and generation tasks, making them an ideal choice for our querying phase.

For specializing the LLM in providing grounded responses, we leveraged the *Auto-Train Advanced*[5] library from HuggingFace. Notably, we employed the PEFT methodology

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[3] https://qdrant.tech/

[4] https://grpc.io/docs/what-is-grpc/introduction/

[5] https://github.com/huggingface/autotrain-advanced

introduced in Section 3 to reduce the computational cost associated with adapting the model to the task at hand, while capitalizing on the main benefits of fine-tuning. Specifically, we fine-tuned the LLaMA 3.1 8B model in a *supervised* fashion using a dataset containing information about a specific BP model (see Section 5 for more details). This dataset included process-specific questions paired with corresponding binary answers about the structural and behavioral information of the process model. By *structural information*, we denote the presence of activities, events, and gateways within the process model, whereas *behavioral information* encompasses details concerning the sequence flows linking these entities.

4.3 RAG querying

This phase includes the following steps: *Retrieval* and *Answering*.

4.3.1 Retrieval

In this stage, the relevant process model chunks useful to generate accurate and contextually relevant answers to the user queries are retrieved. This component has been implemented by integrating the vector store (qDrant) within a retriever component (implemented with an LLM chain with Langchain). The LLM chain serves as the backbone for our retrieval process: it combines retrieval techniques with question-answering capabilities, allowing us to efficiently extract relevant information from process model data. The chain first performs a retrieval step to fetch relevant chunks, then it passes those textual segments into an LLM to generate a response.

To leverage qDrant within the Langchain architecture, a qDrant client needs to be configured as the retriever component. This client interacts with the qDrant vector index, querying it to retrieve process model chunks that are relevant to user queries.

Through experimental analysis, we also determined the number of chunks yielding optimal results. This number was found to strike a balance between providing sufficient context for generating accurate answers and respecting the context window of the language model. Retrieving a sufficient number of process model chunks is fundamental for ensuring that the retrieved content contains all the necessary context for generating accurate answers.

4.3.2 Answering

In the answering step, accurate and contextually relevant answers to user queries are generated. This phase is essential for providing grounded responses based on the input query and the context extracted from relevant process model chunks retrieved in the previous phase, enhancing the overall effectiveness of the BPLLM pipeline.

The answering stage requires two main components: an LLM and the corresponding tokenizer of the model. First, the prompt is crafted by combining the user query with the contextually relevant chunks retrieved in the previous phase as reported in Fig. 4. The tokenizer preprocesses the prompt and converts it into a format that can be understood by the model. Tokenization involves breaking down the input text into individual tokens and encoding them into numerical representations that the model can process. Once the prompt is prepared and tokenized, we feed it into the LLM to generate responses. Relying on the context provided by the retrieved process model chunks, the language model can generate answers that are accurate and contextually relevant to the user's query.

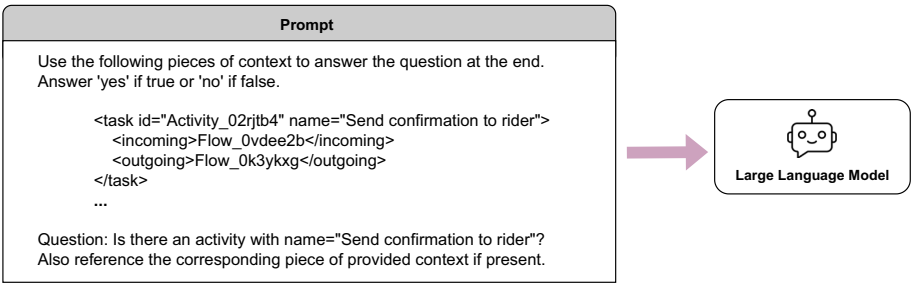

Fig. 4 Example of the prompt passed in input to the LLM

5 Experimental evaluation

This section discusses the experiments conducted to answer the research questions introduced in Section 1. To this end, in the following sub-sections, we describe the process models used for the experiments, the experimental setup, and the results.

5.1 The adopted process models

In this work, we considered three distinct BPs. Specifically, the first process is analyzed using textual DFG descriptions of activities and sequence flows, alongside the corresponding BPMN process model. In contrast, the remaining two processes are evaluated solely through their BPMN models.

We provide a brief overview of the BPs under consideration.

- The initial process, called *Food Delivery*, involves a food delivery system that receives user orders as input and verifies their status as existing clients. If they are not present in the customer database, it creates an account for them. Then, the system checks the specified payment methods, assigns the processed order to an accepting rider for delivery, and communicates the estimated waiting time to the customer. Upon the rider collecting the order, a notification is sent to the client who, upon receipt, notifies the system accordingly to unlock the rider's payment.
- The second process, called *E-commerce*, mirrors the previous one and centers on fulfilling orders placed by customers through an e-commerce platform. Upon receipt of a new order, the system verifies the customer's account status; if not present, it generates one and stores the payment methods. Then, it records the new order, verifies courier availability for delivery, and selects the optimal one. Subsequently, the system assigns the package to the courier and, upon its confirmation, calculates and communicates the delivery date to the customer. After the courier retrieves the package, a confirmation is dispatched to the customer, triggering a reciprocal notification upon order receipt by the customer. Afterward, the customer fills and sends back a satisfaction questionnaire, and the courier is paid.
- The last process, called *Reimbursement*, is entirely distinct from the previous ones and revolves around an expense reimbursement process. It begins with the system's reception of an expense report from an employee, followed by a notification to the employee. In the absence of an existing account, the system creates one for the employee, and a clerk reviews the amount, reformulating it if necessary before submission to the manager. If the amount falls below a specified threshold, it is promptly approved; otherwise, the manager

evaluates it. In case of rejection, the process terminates immediately. If approved, the corresponding amount is transferred to the applicant's bank account, accompanied by a notification.

5.2 Experimental setting

The proposed BPLLM system is based on generative models and hence it generates feedback for the user in natural language. This kind of systems requires a twofold validation based on both *qualitative* and *quantitative* aspects. For this reason, to evaluate the effectiveness of BPLLM in helping users understand BPs, we designed experiments covering these two different perspectives.

More precisely, the quantitative validation covers the correctness (in terms of accuracy) of LLM's answers concerning the entities and relationships in the model provided by the user. Conversely, the qualitative validation is based on the analysis from the user-effectiveness standpoint of experts judging the answers quality.

The conclusion drawn from the study focuses on the overall utility of the BPLLM system in aiding BP users, discussing its potential applications in real-world settings. Based on the outcomes of these experiments, recommendations for the system's improvements are suggested, along with proposals for further studies to continue validating and enhancing the system's capabilities.

Quantitative evaluation

Several experiments are proposed in this section to answer the research questions reported in Section 1. The overall idea is to evaluate the BPLLM performance covering different aspects (RAG, chunking, embedding, and fine-tuning). All the evaluations are performed using the Food Delivery process described in the previous section. The Food Delivery process is represented using the DFG in natural language (in the following we use the term "natural text" for brevity). Figure 5 illustrates the natural language description of the DFG, identifying the process model fed to the LLM.

The queries[6] adopted in these experiments require to be answered in order to recognize *structural* information and *behavioral* aspects within the model. Figure 6 sketches the typology of checks performed on an excerpt of the Food Delivery reference process. For each kind of check (activity and direct flow presence/absence), different prompts are generated to obtain a precise answer from the model.

Specifically, for *structural* information correctness analysis, we queried the presence of specific activities within the BP model, prompting the pipeline to answer with "yes" or "no" and to provide contextual references if available. In evaluating *behavioral* aspects, we formulated questions to determine the existence of sequence flows between activities and asked the LLM to indicate their presence in a binary way along with contextual information if present. We examined all single-pass flows along with an equal number of flows from one existing activity to another but not directly linked in the process and flows between false activities not contained in the model. In both evaluations, we checked existing and non-existing activities and flows and sequence flows for a comprehensive assessment of the framework.

RQ1 assesses the performance (i.e., the accuracy) of BPLLM with respect to the base LLM to address queries concerning the Food Delivery process model.

[6] https://zenodo.org/doi/10.5281/zenodo.13342039

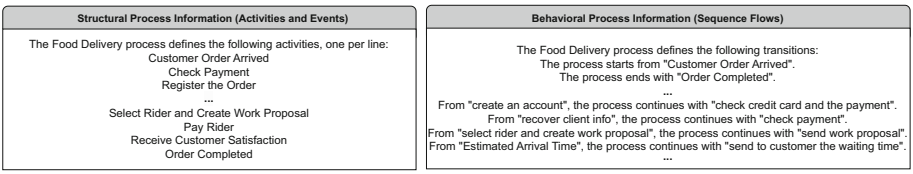

Fig. 5 Extracts of the DFG in natural language provided to the LLM

Indeed, we estimated the effectiveness of the LLM and the BPLLM framework in describing process-related aspects based on *accuracy* metric. To this end, we devised the previously introduced binary methodology to evaluate the pipeline's performance using binary response questions ("yes" or "no"), enabling a rigorous assessment of the responses generated by the LLM . *Accuracy* measures the percentage of exact predictions made by the LLM in addressing such inquiries out of the total number of expected answers. We designate the framework's answers that align with positive expected responses as *true positives* (TP), and those matching negative expected responses as *true negatives* (TN). Conversely, *false positives* (FP) occur when the pipeline yields positive responses contrary to the negative expected ones, while *false negatives* (FN) arise from the framework providing negative responses despite positive expectations. Thus, we defined accuracy as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

RQ2 evaluates the effects of employing various chunking techniques within the BPLLM pipeline, alongside investigating how prompt engineering can further increase the performance of the framework. To this end, different chunking techniques are explored when the

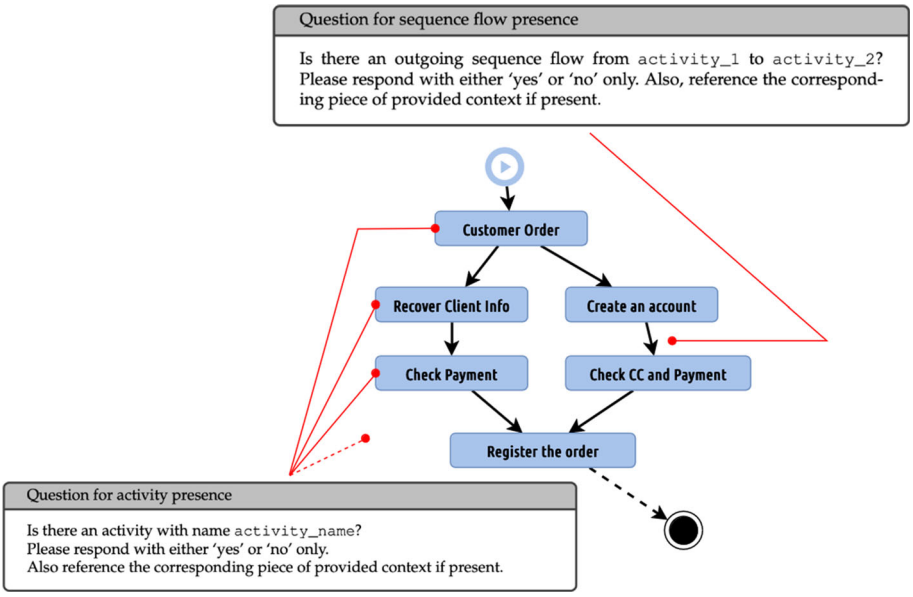

Fig. 6 Prompts for structural and behavioral evaluation

input is in natural text and BPMN format. In the DFG case, fixed-size and recursive chunking with different sizes (16, 64, 128, 256, 512) are tested. For the BPMN format, we estimated also the impact of the customized chunking technique. In both cases, we evaluated the accuracy (see Formula 1) of BPLLM in answering the queries. Additionally, RQ2 evaluates the effects of prompt refinement on the answer's accuracy. For each question, two levels of refinement have been considered (not refined and refined version). The adopted refined and not refined questions are reported in the replication package associated with this paper.

Once we identified the best configuration in terms of accuracy (BPMN for process representation, BPMN-specific chunking, and refined prompt), we conducted further experiments under the same settings employing various open-source language models from the Llama and Mistral families, to extend the evaluation to LLMs with comparable parameters' sizes but larger context windows.

RQ3 examines the impact of employing different representations for the process models within the BPLLM pipeline, as well as whether different embedding models affect our pipeline. In this direction, the natural text and the BPMN model representation for the Food Delivery process are evaluated as described in the previous RQs. For each process representation, different embedding models are tested: *all-MiniLM-L6-v2*[7], which projects sentences into a 384-dimensional dense vector space; *bert-finetuned-bpmn*[8], a fine-tuned version of *bert-base-cased* trained on a dataset comprising textual process descriptions; *paraphrase-xlm-r-multilingual-v1*[9], mapping sentences to a 768-dimensional vector space.

RQ4 explores the effect of fine-tuning the language model in BPLLM coupled with the use of RAG on the accuracy of the framework. To this end, the BPMN representation for the Food Delivery process is evaluated as described in the previous RQs. This experiment tests two variants of fine-tuning, employing quantization with intervals of *int4* and *int8*. *Quantization* consists of reducing the precision of numerical values and is often used to compress LLMs for more efficient deployment. The main difference between *int4* and *int8* quantization lies in the level of precision of the numerical values used to represent the model parameters, with *int8* offering higher precision at increased computational costs.

Finally, the RQ4 also aims to evaluate if the number of processes included in the knowledge base of BPLLM impacts its overall performance. To answer this question, starting from the best BPLLM configuration, different combinations of processes (Food Delivery, Reimbursement, and E-commerce) are included in the initial knowledge base to assess their effect on the BPLLM accuracy.

We conducted the evaluation using an *oracle* designed for this purpose. The oracle receives the question and the expected binary answer as input, compares it with the response generated by the LLM, and calculates the accuracy as the percentage of correct results out of the total number of tests in the specific evaluation.

In our experimentation, we found that by retrieving the top 10 chunks, we could always capture a comprehensive overview of the process models, enabling the language model to generate informed responses.

The experiments were conducted on a workstation equipped with Linux/Ubuntu 22.04.3 LTS operating system and powered by an NVIDIA A100 GPU.

[7] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[8] https://huggingface.co/jtlicardo/bert-finetuned-bpmn

[9] https://huggingface.co/sentence-transformers/paraphrase-xlm-r-multilingual-v1

5.2.1 Qualitative evaluation

Sticking to the optimal BPLLM configuration, we conducted an additional qualitative assessment of the pipeline by posing more nuanced inquiries about the whole BPMN process model. This was undertaken to simulate real-world users' interactions with the tool, aimed at gaining deeper insights into the underlying BP model or addressing specific issues related to their work.

We centered the qualitative evaluation on the Food Delivery process, providing its entire BPMN model as context for the framework (removing only graphical information, deemed irrelevant for this analysis).

5.3 Evaluation results

We proceed to analyze the results obtained during the evaluation phase under various experimental conditions for each RQ introduced in Section 1.

RQ1: ***How does the adoption of Retrieval-Augmented Generation (RAG), in the frame of BPLLM architecture, impacts the end-to-end performance?***

Table 1 reports the accuracy obtained when a pure LLM and BPLLM are used on the Food Delivery model described in natural text and BPMN. To adhere to the context window of the LLM, we removed the graphical information from the BPMN file, as it was irrelevant to our objectives. The table shows a significant enhancement in performance when BPLLM is used, aligning with our expectations. This improvement shows better accuracy for the RAG-based LLM leveraging the natural language representation to drive more informed and accurate decision-making.

Additionally, our observations revealed instances of hallucination, wherein the pure LLM responds by lacking information about the process model, occasionally asserting familiarity with certain activities even in the absence of such knowledge.

RQ2: ***What's the influence of chunking and prompting on the performance of the BPLLM?***

Table 2 reports the accuracy evaluated on different process model representations (*natural text* and *BPMN*) and distinct chunking techniques (*no chunking*, *fixed-size*, *recursive*, *BPMN-specific*).

When natural text is used, we obtained comparable results by relying on fixed chunking and recursive chunking. In both scenarios, the optimal chunk size is determined to be 128 tokens with a 10-tokens overlap. This finding can be attributed to the process model's small size, whose content is nearly included within a single chunk. This also explains why the case of "no chunking" yields similar results.

Employing the BPMN format, the results are quite different. Favorable outcomes are obtained when fixed chunking and recursive chunking are used. The optimal chunk size in both cases is 32 tokens with a 10-token overlap. Nonetheless, a great difference in accuracy can be observed when different process models are used. This is in line with the idea that the

Table 1 Comparison of the evaluation results for pure LLM and RAG framework

Methodology	Process 1	Language model	Accuracy
Pure LLM	None	Llama 2 13B	42.78%
RAG	Natural Text	Llama 2 13B	**65.48**%
RAG	BPMN	Llama 2 13B	60.57%

The best result is highlighted in bold

Table 2 Evaluation results after chunking and prompts refinements

Process Model Representation	Chunking	Prompts	Language Model	Accuracy
Natural Text	No Chunking	Not Refined	Llama 2 13B	65.48%
Natural Text	No Chunking	Refined	Llama 2 13B	67.27%
Natural Text	Fixed	Not Refined	Llama 2 13B	54.76%
Natural Text	Fixed	Refined	Llama 2 13B	60.71%
Natural Text	Recursive	Not Refined	Llama 2 13B	55.95%
Natural Text	Recursive	Refined	Llama 2 13B	61.90%
BPMN	No Chunking	Not Refined	Llama 2 13B	60.57%
BPMN	No Chunking	Refined	Llama 2 13B	64.86%
BPMN	Fixed	Not Refined	Llama 2 13B	60.78%
BPMN	Fixed	Refined	Llama 2 13B	73.53%
BPMN	Recursive	Not Refined	Llama 2 13B	67.65%
BPMN	Recursive	Refined	Llama 2 13B	73.53%
BPMN	BPMN-specific	Not Refined	Llama 2 13B	68.63%
BPMN	BPMN-specific	Refined	Llama 2 13B	76.47%
BPMN	BPMN-specific	Refined	Llama 2 7B	68.47%
BPMN	BPMN-specific	Refined	Llama 3 8B Instruct	89.78%
BPMN	BPMN-specific	Refined	Llama 3.1 8B Instruct	**90.38**%
BPMN	BPMN-specific	Refined	Mistral Instruct 7B v0.2	88.85%
BPMN	BPMN-specific	Refined	Mistral Instruct 7B v0.3	90.18%
BPMN	BPMN-specific	Refined	Mixtral 8x7B Instruct v0.1	79.81%

The best result is highlighted in bold

process model size influences the chunking strategy since the BPMN model is greater than the natural text model in size.

The best results for Llama 2 13B are obtained by applying the BPMN-specific chunking, showing an accuracy of *76.47 percent*. This confirms the idea that a custom technique for chunking BPMN representations into semantically valid segments resulted in a significant enhancement in accuracy for BPLLM, underscoring the effectiveness of chunking strategies tailored to the specific process model representation.

Furthermore, Table 2 reports, in the third column, the refinement level of the adopted prompt (*not refined* and *refined*). In all the cases, the refined prompts demonstrated greater accuracy in the answer. For the BPMN representation, the difference in accuracy between the not refined and refined prompt cases is significant, exhibiting a strong influence of the prompt refinement level on the results.

Based on the optimal configuration identified with these results, we experimented with the language model's choice. First, we tested Llama 2 7B to determine if the *inverse scaling* phenomenon applied in this context (McKenzie et al., 2023). Indeed, although it is generally expected that increasing the model's scale leads to better performance, this is not always the case, as sometimes larger LLMs perform worse on specific tasks. However, our experiments with Llama 2 7B did not produce favorable outcomes. Thus, we shifted to more advanced models with larger context windows from the Llama 3 and Mistral families. Ultimately, Llama 3.1 8B Instruct[10] delivered the best results, yielding an accuracy of *90.38 percent*.

[10] https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct

RQ3: ***How does the choice of process representation format and related embedding model affect the performance of the BPLLM?***

In tackling RQ3, we considered the impact of the process representation format and embedding model on BPLLM accuracy. Table 3 summarizes the collected insights.

Specifically, we observed satisfactory outcomes with the natural language representation. In this case, optimal performance (*87,88 percent*) is attained when employing the embedding model tailored for BPMN terminology, namely *bert-finetuned-bpmn*. Conversely, for the BPMN process model, superior results (*90,38 percent*) are achieved by leveraging the *all-MiniLM-L6-v2* model. Additionally, we evaluated the *paraphrase-xlm-r-multilingual-v1* model. It demonstrated decent performance with both the process representations but with no significant improvements in accuracy compared to the other embedding models.

Based on these findings, we generated t-SNE charts for the most effective combinations, as depicted in Fig. 7. For this experiment, we considered the entire prompt containing the process-related query and the relevant context retrieved from the vector database based on cosine similarity. In the left chart, illustrating the embeddings derived from the *bert-finetuned-bpmn* model, noticeable proximity between the prompts and the chunks of the natural text model can be observed, thus confirming the reliability of the results. Similarly, the t-SNE on the right illustrates that the embeddings produced by the *all-MiniLM-L6-v2* model are more suitable for the BPMN representation of the process model, as both the prompts and the BPMN chunks are concentrated within the same area.

RQ4: ***What is the effect of fine-tuning along with the use of RAG on the performance of the BPLLM? Does the number of processes included in the knowledge base of a single LLM impact performance?***

The outcomes of the experiments directed at evaluating the influence of fine-tuning on the performance of the BPLLM are reported in Table 4. The table shows the BPLLM accuracy for different PEFT quantizations (*int4* and *int8*). We considered in the table, for brevity, the best configuration of parameters according to the results described in the other subsections (i.e., *BPMN model* as input, *BPMN-specific chunking*, *all-MiniLM-L6-v2* as embedding model, and *Llama 3.1 8B* as language model). In the first two rows, the table shows the results when Llama 3.1 8B is fine-tuned only on the Food Delivery process. In this case, the *int4* quantization demonstrated superior performance (*92,31 percent*) not only in terms of reduced computational costs but also in the final accuracy of the BPLLM. The comparison of Table 4 and the best case reported in Table 3 shows increased accuracy when fine-tuning is performed.

Table 3 Evaluation results for different representations and embedding models

Process Model Representation	Embedding Model	Language Model	Accuracy
Natural Text	all-MiniLM-L6-v2	Llama 3.1 8B Instruct	87.50%
Natural Text	paraphrase-xlm-r-multilingual-v1	Llama 3.1 8B Instruct	85.58%
Natural Text	bert-finetuned-bpmn	Llama 3.1 8B Instruct	87.88%
BPMN	all-MiniLM-L6-v2	Llama 3.1 8B Instruct	**90.38%**
BPMN	paraphrase-xlm-r-multilingual-v1	Llama 3.1 8B Instruct	90.20%
BPMN	bert-finetuned-bpmn	Llama 3.1 8B Instruct	69.23%

The best result is highlighted in bold

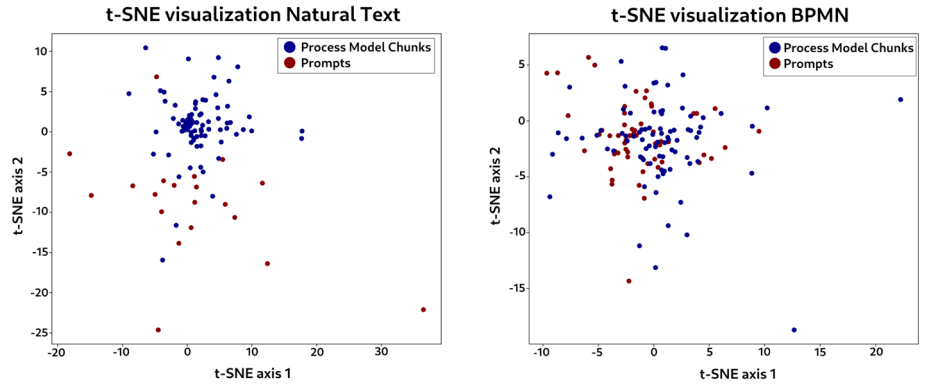

Fig. 7 T-SNE charts depicting the embeddings of the prompts and chunks derived from the natural text (left) and the BPMN (right) process model. The embeddings are computed using bert-finetuned-bpmn (left) and all-MiniLM-L6-v2 (right)

Continuing our analysis of the results, we examined the potential effects of enlarging the knowledge base within the vector database by incorporating multiple process models to provide to BPLLM. Our investigation considered both process models having similar and different activities and objectives to assess the performance implications on the framework. By similar activities, we refer to tasks that have synonymous names and share similar objectives (e.g., "*Select Rider*" and "*Select Courier*"). In contrast, different activities pertain to tasks with distinct names and actions (e.g., "Estimate Arrival Time" and "*Approve Expense Report*").

Initially, we examined the impact of integrating two distinct BPs (namely the previously introduced Food Delivery and Reimbursement process models) into the knowledge base. Our findings indicated a slight degradation in the performance of the BPLLM pipeline. However, fine-tuning the LLM with *int4* quantization on both models mitigated the adverse effects, demonstrating improved accuracy. Conversely, when incorporating two similar BP models (such as Food Delivery and E-commerce), we observed a marked decline in accuracy for both the base BPLLM and the pipeline utilizing the fine-tuned LLM. This outcome was expected, as the LLM can be influenced in decision-making by the similarities in activity names between the BP models, leading to potential misinterpretations.

Table 4 Evaluation results for different PEFT quantization and enlarged knowledge base

Process Model(s)	Fine-Tuning	PEFT Quantization	Accuracy
Food Delivery	✓	int4	**92.31%**
Food Delivery	✓	int8	90.58%
Food Delivery, Reimbursement		–	82.16%
Food Delivery, Reimbursement	✓	int4	84.62%
Food Delivery, Reimbursement	✓	int8	81.87%
Food Delivery, E-commerce		–	80.26%
Food Delivery, E-commerce	✓	int4	80.53%
Food Delivery, E-commerce	✓	int8	79.85%

The best result is highlighted in bold

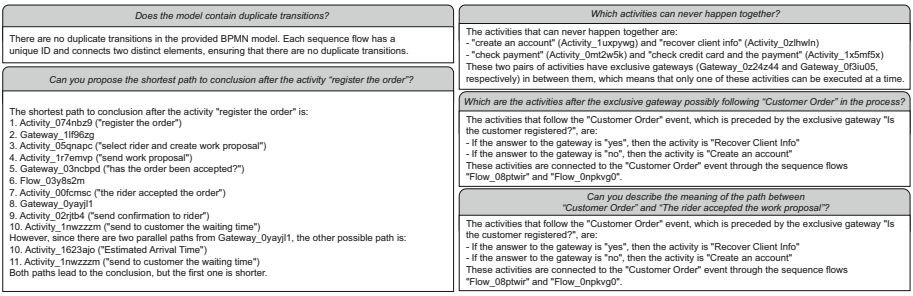

Fig. 8 An extract of the qualitative evaluation results

These findings underscore the importance of carefully managing the size and composition of the knowledge base within BPLLM, particularly when incorporating multiple process models with varying degrees of similarity. Additionally, our results highlight the potential benefits of fine-tuning the LLM to enhance its adaptability and performance across diverse BP contexts.

5.3.1 Qualitative evaluation discussion

We also evaluated the capability of BPLLM to respond appropriately to nuanced process-wide inquiries posed by a group of potential users, considering the whole Food Delivery BPMN model in input. The complete set of questions and answers can be found in the replication package[11]. Figure 8 illustrates an informative sample for this experimentation. Notably, it shows the ability of BPLLM to: (*i*) identify duplicate elements in a process model, ensuring process integrity and correctness; (*ii*) navigate and optimize process paths for efficiency analysis; (*iii*) understand exclusive gateways and mutually exclusive activities; (*iv*) trace process flows; and (*v*) explain sequences of events, demonstrating comprehension of the rationale behind BP steps.

The selected users, based on the obtained answers, posit that the framework exhibits the ability to provide satisfactory responses promptly, thereby establishing its reliability for everyday usage.

6 Threats to validity

This section reports some considerations about the threats to the validity of the study. A first threat to internal validity concerns the choice to use in our BPLLM approach open LLM models since there are closed models (like ChatGPT) largely recognized by the community that can give good results. Regarding to this point, we consider that in a research setting the adoption of a closed model is impractical due to flexibility issues, the high costs of these solutions, and the possibility to control and evaluate the development process. Referring to the choice of adopting open-source models, another internal validity concern refers to the rapid advancement of LLM models (new versions are developed in a few months) that makes the obtained results outdated in time. To counter this, in the proposed BPLLM, the more recent open LLM models of the LLaMA family are evaluated. However, the considered BPLLM solution can be easily adapted to new overruling LLaMA models.

[11] https://zenodo.org/doi/10.5281/zenodo.13342039

Instead, an external validity threat to this study is given by the difficulty in replicating the proposed experiments since LLMs can provide variable responses to the same prompt. To mitigate this critical issue, we made available online the adopted prompts and evaluated as possible answers only binary responses ("yes" or "not"). This makes the experiments more reproducible and reduces the differences in how questions or commands are phrased. Moreover, we performed multiple runs along the experiment to ensure the correctness of the answer. Finally, always referring to the external validity, we focused on ensuring the generalizability of our results by extending the experimentation to a larger number of process models.

7 Conclusion

In this paper, we introduced BPLLM, a novel methodology aimed at enabling grounded conversations with an LLM to enhance process-aware DSSs. The framework is designed for the analysis and description of BPs, boosting the conversational capabilities of LLMs in addressing process-related tasks. It achieves this by integrating a RAG framework to ingest contextual knowledge relevant to specific user queries with fine-tuning, thus extending process-specific structural and behavioral knowledge. In this way, BPLLM fine-tuned on a designated process model can assist users in various process comprehension and execution tasks through natural language interactions. Furthermore, we introduced a process-aware chunking approach and evaluated the BPLLM pipeline using different embedding models to identify the most suitable ones. Additionally, we proposed fine-tuning the LLM model to enhance the understanding of process definitions. We assess BPLLM's capability to generate accurate and contextually relevant responses to user queries about processes. Lastly, we evaluated the proposed approach across various BP models, examining its performance with an expanded knowledge base.

As part of our future work in process discovery (Bernardi et al., 2014), we aim to include an analysis of the execution aspects of BPs in the study. This study will involve considering data about the execution times and costs of activities, enabling us to offer valuable insights into the process execution to users. Additionally, we plan to explore various embedding models to enable the computation of proximity or distance between distinct execution traces within an event log generated during the enactment of a BP. Future research directions also encompass enhancing the retrieval phase by investigating the integration of knowledge graphs into our approach. Ultimately, exploring the combination of the BPLLM framework with symbolic AI solvers may represent another interesting future avenue, aiming to incorporate aspects of reasoning into the methodology.

Acknowledgements The work of A. Casciani is in the range of the Italian National Doctorate on AI run by Sapienza. This work is supported by the Sapienza project FOND-AIBPM, the PRIN 2022 project MOTOWN and the PNRR MUR project PE0000013-FAIR.

Author Contributions A. Casciani carried out the experiments. M.L. Bernardi, A. Casciani, and M. Cimitile wrote the manuscript. All authors reviewed the manuscript.

Funding Open access funding provided by Universitá degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

Data Availability Replication package at: https://zenodo.org/doi/10.5281/zenodo.13342039.

Declarations

Competing Interests The authors declare no competing interests.

Ethical Approval Not Applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

van der Aa, H., Carmona, J., Leopold, H. et al (2018) Challenges and opportunities of applying natural language processing in business process management. In: COLING 2018: The 27th Int. Conf. on Computational Linguistics, pp. 2791–2801

van der Aalst WMP (2019) A practitioner's guide to process mining: Limitations of the directly-follows graph. In: CENTERIS 2019 - Int. Conf. on ENTERprise Inf. Sys. / ProjMAN 2019 - Int. Conf. on Project MANagement / HCist 2019 - Int. Conf. on Health and Social Care Inf. Sys. and Tech. 2019, pp. 321–328. https://doi.org/10.1016/J.PROCS.2019.12.189

Agarwal, P., Gao, B. et al (2022) A Process-Aware Decision Support System for Business Processes. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, *Proceedings*, p. 2673–2681. https://doi.org/10.1145/3534678.3539088

Agostinelli, S., De Luzi, F., Di Canito, U. et al (2022) A data-centric approach to design resilient-aware process models in BPMN. In: Business Process Management Forum - BPM 2022 Forum, *Proceedings*, Springer, pp. 38–54. https://doi.org/10.1007/978-3-031-16171-1_3

Ali, R., Hussain, A., Nazir, S. et al (2023) Intelligent Decision Support Systems: An Analysis of Machine Learning and Multicriteria Decision-Making Methods. *Applied Sciences 13*(22) https://doi.org/10.3390/app132212426

Bennett, C. C., & Hauser, K. (2013). Artificial intelligence framework for simulating clinical decision-making: A Markov decision process approach. *Artif Intell Med, 57*(1), 9–1. https://doi.org/10.1016/j.artmed.2012.12.003

Bernardi, ML., Cimitile, M., Di Francescomarino, C. et al (2014) Using discriminative rule mining to discover declarative process models with non-atomic activities. In: Rules on the Web. - RuleML 2014, ECAI 2014. *Proceedings*, pp. 281–295

Berti, A., Schuster, D., van der Aalst, WMP. (2023) Abstractions, Scenarios, and Prompt Definitions for Process Mining with LLMs: A Case Study. In: Business Process Management Workshops - BPM 2023 International Workshops, pp. 427–439. https://doi.org/10.1007/978-3-031-50974-2_32

Casciani, A., Bernardi, ML., Cimitile, M. et al. (2024) Conversational Systems for AI-Augmented Business Process Management. In: Research Challenges in Information Science RCIS 2024, *Proceedings*, Part I, pp. 183–200. https://doi.org/10.1007/978-3-031-59465-6_12

Ceravolo, P., Junior, S. B., Damiani, E., et al. (2024). Tuning machine learning to address process mining requirements. *IEEE Access, 12*, 24583–24595. https://doi.org/10.1109/ACCESS.2024.3361650

Chapela-Campa, D., Dumas, M. (2023). From process mining to augmented process execution. *Software and Systems Modeling 22*(6) 1977–1986. https://doi.org/10.1007/s10270-023-01132-2

Conforti, R., de Leoni, M., Rosa, ML. et al. (2015) A recommendation system for predicting risks across multiple business process instances. *Decis Support Syst* 69 1–19. https://api.semanticscholar.org/CorpusID:8864943

Dong, Q., Li, L., Dai, D. et al. (2024) A survey on in-context learning. arXiv:2301.00234

Dubey, A., Jauhri, A., Pandey, A. et al. (2024). The Llama 3 Herd of Models. arXiv:2407.21783

Dumas, M., La Rosa, M., Mendling, J., et al. (2013). *Fundamentals of Business Process Management*. Springer.

Dumas, M., Fournier, F., Limonad, L. et al. (2023). Ai-augmented business process management systems: A research manifesto. *ACM Trans Manag Inf Syst 14*(1) 11:1–11:19. https://doi.org/10.1145/3576047

Estrada-Torres, B., del Río-Ortega, A., Resinas, M. (2024) Mapping the landscape: Exploring large language model applications in business process management. In: Enterprise, Business-Process and Inf. Sys. Modeling. Springer Nature, pp. 22–31

Fahland, D., Fournier, F., Limonad, L., et al. (2024) How well can large language models explain business processes? CoRR. https://doi.org/10.48550/ARXIV.2401.12846

Fontenla-Seco, Y., Lama, M., Bugarín, A. (2020) Process-To-Text: A Framework for the Quantitative Description of Processes in Natural Language. In: Trustworthy AI - Integrating Learning, Optimization and Reasoning - Workshop, TAILOR 2020, LNCS, vol. 12641. Springer, pp. 212–219

Gao, Y., Xiong, Y., Gao, X., et al. (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997

Grohs, M., Abb, L., Elsayed, N. et al. (2024). Large language models can accomplish business process management tasks. In: Business Process Management Workshops. Springer Nature, pp. 453–465

Hosseini, M. T., Ghaffari, A., Tahaei, M. S., et al. (2023). Towards Fine-tuning Pre-trained Language Models with Integer Forward and Backward Propagation. *Findings of the Association for Computational Linguistics: EACL, 2023*, 1867–1876.

Jiang, AQ., Sablayrolles, A., Mensch, A., et al. (2023). Mistral 7b. CoRR https://doi.org/10.48550/ARXIV.2310.06825

Jiang, AQ., Sablayrolles, A., Roux, A., et al. (2024). Mixtral of experts. CoRR. https://doi.org/10.48550/ARXIV.2401.04088

Lashkevich, K., Milani, F., Avramenko, M., et al. (2024). Llm-assisted optimization of waiting time in business processes: A prompting method. In: Business Process Management. Springer Nature, pp. 474–492

Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Int. Conf. on Neural Information Processing Systems, Proceedings, NIPS'20

Lohrmann, M., & Reichert, M. (2016). Effective application of process improvement patterns to business processes. *Software & Systems Modeling, 15*(2), 353–375. https://doi.org/10.1007/s10270-014-0443-z

López, A., Sànchez-Ferreres, J., Carmona, J., et al. (2019). From Process Models to Chatbots. In: Int. Conf. on Advanced Information Systems Engineering, https://api.semanticscholar.org/CorpusID:169031222

McKenzie, IR., Lyzhov, A., Pieler, M., et al. (2023). Inverse Scaling: When Bigger Isn't Better. *Trans Mach Learn Res* 2023

Mozannar, H., Sontag, DA., (2020). Consistent estimators for learning to defer to an expert. In: Proceedings of the 37th Int. Conf. on Machine Learning, ICML 2020, *Proceedings of Machine Learning Research*, vol 119. PMLR, pp. 7076–7087

Naveed, H., Khan, AU., Qiu, S., et al. (2023). A comprehensive overview of large language models. CoRR. https://doi.org/10.48550/ARXIV.2307.06435

Object Management Group (2011) Business Process Model and Notation (BPMN), Version 2.0. http://www.omg.org/spec/BPMN/2.0

Ozkaya, I. (2023). Application of Large Language Models to Software Engineering Tasks: Opportunities, Risks, and Implications. *IEEE Software, 40*(3), 4–8. https://doi.org/10.1109/MS.2023.3248401

Pasquadibisceglie, V., Appice, A., Malerba, D., (2024). Lupin: A llm approach for activity suffix prediction in business process event logs. In: 2024 6th Int. Conf. on Process Mining (ICPM), pp 1–8, https://doi.org/10.1109/ICPM63005.2024.10680620

Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems, 3*, 121–154.

Schonenberg, H., Weber, B., van Dongen, B., et al. (2008). Supporting Flexible Processes through Recommendations Based on History. *Business Process Management* (pp. 51–66). Berlin Heidelberg, Berlin, Heidelberg: Springer.

Touvron, H., Lavril, T., Izacard, G., et al. (2023a) Llama: Open and efficient foundation language models. CoRR. https://doi.org/10.48550/ARXIV.2302.13971

Touvron, H., Martin, L., Stone, K., et al. (2023b) Llama 2: Open foundation and fine-tuned chat models. CoRR. https://doi.org/10.48550/ARXIV.2307.09288

Vaisman, A. (2013). An introduction to business process modeling. In: Business Intelligence: Second European Summer School, eBISS 2012. Springer, pp. 29–61

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems 30*

Voorberg, S., Eshuis, R., van Jaarsveld, W., et al. (2019) Decision Support for Declarative Artifact-Centric Process Models. In: Business Process Management Forum. *Springer International Publishing*, pp. 36–52

Xu, L., Wang, W. (2023). Improving aspect-based sentiment analysis with contrastive learning. *Natural Language Processing Journal, 3*,

Xu, L., Xie, H., Li, Z., et al. (2023a) Contrastive learning models for sentence representations. *ACM Transactions on Intelligent Systems and Technology 14*(4) 1–34

Xu, L., Xie, H., Qin, SJ., et al. (2023b) Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. CoRR. https://doi.org/10.48550/ARXIV.2312.12148
Zhao, WX., Zhou, K., Li, J., et al. (2023). A survey of large language models. CoRR. https://doi.org/10.48550/ARXIV.2303.18223

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.