Casciani *et al. Process Science* (2026) 3:5
https://doi.org/10.1007/s44311-026-00037-9

Process Science

RESEARCH Open Access

# Enabling natural language analysis for object-centric event logs

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

*Correspondence:
Angelo Casciani
casciani@diag.uniroma1.it
[1]Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Rome 00185, Italy
[2]Department of Engineering, University of Sannio, Piazza Roma 21, Benevento 82100, Italy
[3]Department of Law and Digital Society, Unitelma Sapienza, Piazza Sassari 4, Rome 00185, Italy

**Abstract**

Object-centric process mining is an emerging family of techniques for analyzing business processes where events involve multiple interconnected objects. However, formulating queries for object-centric event logs might be challenging when using traditional query languages such as SQL, as these languages typically demand the user to have programming expertise for handling the intricate relationships inherent in object-centric data. To address this challenge, we introduce a conversational framework designed to facilitate the analysis of object-centric event logs following the OCEL 2.0 format. By leveraging the capabilities of Large Language Models (LLMs) with Retrieval Augmented Generation (RAG), our framework enables users to interact with object-centric logs in natural language, eliminating the need for conventional query languages. Additionally, we present a dataset derived from a SAP Procure-to-Pay event log as a benchmark for assessing the performance of conversational techniques in analyzing object-centric event logs from various perspectives. The evaluation demonstrates that our framework enhances the accessibility of OCEL 2.0 log analysis, providing accurate and faithful responses.

**Keywords** OCEL, Object-centric event log, Large language model, Retrieval-augmented generation

## Introduction

Industry 4.0 has significantly increased the availability of event logs tracing the execution of business processes, making it essential to analyze and optimize these processes to maintain organizational competitiveness effectively. *Process mining*, a field dedicated to discovering, monitoring, and improving real processes by extracting knowledge from event logs, plays a fundamental role in this context (van der Aalst 2022). However, the complexity inherent in process mining techniques often places meaningful insights out of reach for non-technical users (Barbieri et al. 2024). These issues are particularly noticeable for the recently introduced object-centric event logs, which provide a more detailed and interconnected representation of business process execution compared to traditional case-centric event logs (van der Aalst 2023). By capturing interactions among multiple objects involved in processes, object-centric logs offer richer descriptions of process flows. Despite their advantages, analyzing such logs remains difficult, especially



Springer

for users lacking specialized programming expertise. Even general-purpose querying languages such as SQL struggle with expressing queries over object-centric event logs due to their inherent complexity, often requiring costly operations in terms of performance when handling the multi-dimensional structures typical of such data.

Lately, Large Language Models (LLMs) have emerged as a disruptive technology in Natural Language Processing (NLP), capable of understanding and generating human-like text (Naveed et al. 2025). Leveraging their capabilities, conversational interfaces built on them can interpret and respond to natural language queries, thereby democratizing object-centric process mining insights to a broader audience.

In this paper, we introduce a novel *conversational framework* designed to facilitate the analysis of object-centric event logs following the OCEL 2.0 format. Our architecture combines LLMs with *Retrieval Augmented Generation* (RAG) to handle users' queries about logs and generate faithful and contextually relevant responses in natural language. To the best of our knowledge, this is the first structured LLM-based research framework that enables conversational interrogation of object-centric event logs, extracting multi-dimensional information while preserving the interrelations between objects and events. While conversational assistants for process mining are starting to appear in commercial ecosystems, they typically function as proprietary black boxes and are tightly bound to vendor-specific infrastructures. By contrast, our framework emphasizes *openness*, *reproducibility*, and *transparency*.

Indeed, the framework can run *entirely locally* using open-source LLMs, allowing organizations to retain full control over how logs are processed and what information is stored in the local database, an essential requirement in privacy-sensitive domains.

The framework's primary aim is to lower the entry barrier for *non-technical users*. Its target users consist primarily of process owners and managers who possess domain knowledge but lack technical skills in SQL or other formal languages. The framework enables them to ask exploratory questions about execution data, validate preliminary hypotheses, and extract insights that can guide more advanced analyses by technical data analysts.

Regarding its positioning, the framework is designed to *complement* rather than replace traditional process mining analysis. It can also be integrated into broader process mining solutions and ecosystems, including those offered by commercial vendors.

We now introduce a motivating example for our framework. Consider an object-centric event log describing the execution of a *Procure-to-Pay* business process, and suppose we are interested in the query: *"Which material had the highest ordered quantity, and in which event was this recorded?"* Answering the latter requires identifying both the material with the maximum quantity and the precise event in which it occurred. Such a query could be expressed in SQL over the flattened event log using the material as the case identifier as follows:

```
SELECT material_id, MAX(quantity)
FROM log_flattened
GROUP BY material_id;
```

This retrieves the maximum quantity for each material, but it does not return the event in which such a maximum quantity occurred. To obtain that information, additional joins or nested subqueries are required, which substantially increase query complexity.

Even for this apparently simple information, SQL queries quickly become non-trivial once the log is flattened, raising the entry barrier for non-technical users.

Beyond the query complexity, flattening also introduces *semantic distortions* on the event log originating from two well-known issues, i.e., convergence and divergence. *Convergence* occurs when a single event is related to multiple cases, forcing the need to duplicate the event across different instances. For example, consider an event $e_1$ that records the creation of the purchase order $p_1$. Such a purchase order includes materials $m_a$ (with quantity $q = 100$) and $m_b$ (with quantity $q = 50$). Assuming that the event log is flattened using the material as the case identifier, $e_1$ gets duplicated, with an occurrence for the case of $m_a$ and one for the case of $m_b$.

On the other hand, *divergence* arises when events within a single case cannot be properly distinguished, particularly when multiple objects are involved in a single process instance. For instance, suppose flattening the event log on purchase orders and consider a purchase order $p_1$ containing material $m_a$, with quantity $q = 100$, and $m_b$, with $q = 50$. In this scenario, all material objects are treated within the same case, making it harder to distinguish between the lifecycles of material $m_a$ and $m_b$.

In contrast, the proposed framework can accurately identify the maximum quantity ordered for a material object and link it to the specific event by leveraging the preserved event-object relationships and object attributes, providing a grounded answer and avoiding the issues caused by flattening the event log.

This paper tackles the following research questions (RQs):

**RQ1**: *How accurate are the framework's responses in supporting object-centric event log analysis?*

**RQ2**: *To what extent is the conversational interface beneficial for users without technical expertise?*

To address these questions, we conducted a twofold evaluation. For **RQ1**, we performed a structured *quantitative analysis* to measure the framework's performance. This involved an ablation study across a range of LLMs and prompt engineering strategies using standard classification metrics. For **RQ2**, we conducted a *qualitative user study* using the best framework's configuration identified in our quantitative assessment to evaluate the clarity, usefulness, and overall benefit of the conversational interface for non-technical users.

Additionally, we present a *dataset* for evaluating our framework, derived from an OCEL 2.0 event log. This dataset functions as a benchmark for evaluating the effectiveness of conversational techniques in analyzing such an event log from multiple perspectives. The evaluation of the framework shows promising results, demonstrating that our framework enhances the interpretability of object-centric event logs through a natural language interaction while maintaining high faithfulness and reliability in its responses.

The rest of the paper is structured as follows. Section 2 provides the necessary background to understand the context and the framework. Section 3 reviews related work in the field, and Sect. 4 describes the conversational framework. Section 5 outlines the conducted experiments and discusses the results. Section 6 reports the validity threats for the study. Finally, Sect. 7 summarizes the contributions and limitations of the work.

## Background

This section provides the necessary background on object-centric event logs in process mining and LLMs.

### Event logs in process mining

Process mining relies on *event logs*, which capture the execution of business processes in terms of events and their attributes (e.g., activity name, timestamp, resource). The traditional paradigm is *case-centric*, where a single case notion (such as an order, claim, or patient) is selected to define process instances. This view is exemplified by the widely adopted XES standard[1] (Acampora et al. 2017).

While effective in many scenarios, the case-centric perspective becomes *problematic* when multiple interacting objects coexist. Flattening event data into a single case notion presents *negative side effects*, obscuring true relationships between objects and events and causing issues such as the *convergence* and *divergence*. Precisely, in the convergence phenomenon, events involving multiple objects of the type selected as the case notion are replicated, possibly leading to unintentional duplication. On the other hand, in the divergence issue, events referring to different objects of a type not selected as the case notion may still be considered causally related because a more coarse-grained object is shared. Mitigating the above two problems is nowadays one of the relevant challenges investigated in the object-centric process mining literature to enable the analysis of data-rich processes (Rebmann et al. 2022; Berti et al. 2025).

To address these limitations, several object-centric event log formats have been proposed, each with its own advantages and disadvantages, but no definitive standard has yet emerged. Examples include *eXtensible Object-Centric logs* (XOC) (Li et al. 2018), *Data-aware Object-Centric Event Logs* (DOCEL) (Goossens et al. 2022), *Artifact-Centric Event Logs* (ACEL) (Moctar-M'Baba et al. 2023), *Event Knowledge Graphs* (EKG) (Fahland 2022), and *Object-Centric Event Log* (OCEL) (Ghahfarokhi et al. 2021).

According to a recent comparative study carried out by Goossens et al. (2024), *OCEL* emerges as one of the most mature formats overall, as it supports a wide range of desired features. For this reason, we adopt its evolution, i.e., OCEL 2.0, as the reference format for event logs in this paper.

### Object-centric event logs (OCEL)

The first version of the *Object-Centric Event Log* (OCEL) format (Ghahfarokhi et al. 2021) was introduced to generalize the case-centric view by allowing events to reference multiple objects of different types (e.g., items, orders, payments, etc.). This enables all relevant events to be included without enforcing a single case notion, relationships between objects to be inferred via shared events, and misleading dependencies from flattening to be avoided.

Formally, an OCEL log can be represented as a tuple:

$$\mathcal{L} = \langle \mathcal{E}, \mathcal{O}, \mathcal{EA}, \mathcal{OA}, \mathrm{act}, \mathrm{time}, \mathrm{objtype}, \mathrm{eaval}, \mathrm{oaval}, \mathcal{E}2\mathcal{O} \rangle$$

where:

- $\mathcal{E} \subseteq \mathcal{U}_{ev}$ is the set of events and $\mathcal{O} \subseteq \mathcal{U}_{obj}$ the set of objects.

[1] IEEE 1849–2023 XES Standard: https://xes-standard.org/

- $\mathcal{EA}$ and $\mathcal{OA}$ are the sets of event and object attributes, respectively.
- $\text{act} : \mathcal{E} \to \mathcal{U}_{act}$ assigns activities to events.
- $\text{time} : \mathcal{E} \to \mathcal{U}_{time}$ assigns timestamps to events.
- $\text{objtype} : \mathcal{O} \to \mathcal{U}_{otype}$ assigns object types to objects.
- $\text{eaval} : (\mathcal{E} \times \mathcal{EA}) \mapsto \mathcal{U}_{val}$ assigns attribute values to events.
- $\text{oaval} : (\mathcal{O} \times \mathcal{OA}) \mapsto \mathcal{U}_{val}$ assigns attribute values to objects.
- $\mathcal{E2O} \subseteq \mathcal{E} \times \mathcal{O}$ relates events to the objects they reference.

Building on this foundation, *OCEL 2.0* (Berti et al. 2024a) introduced several extensions to better capture relationships within event data:

- *qualified event-to-object relationships*, where events reference objects with explicit roles (e.g., an order as container and an item as content in an *"add item"* event);
- *qualified object-to-object relationships*, where direct relations such as *"contains"* can be modeled between objects;
- *temporal evolution of object attributes*, recording how object states change over time (e.g., an item status evolving from *"pending"* to *"shipped"*).

More formally, an OCEL 2.0 log, whose metamodel from the official specification is reported in Fig. 1, is described as:

$$\mathcal{L} = \langle \mathcal{E}, \mathcal{O}, \mathcal{EA}, \mathcal{OA}, evtype, time, objtype, eatype, oatype, eaval, oaval, \mathcal{E2O}, \mathcal{O2O} \rangle$$

where, in addition to the components already defined for the first version of the format:

- $\text{evtype} : \mathcal{E} \to \mathcal{U}_{etype}$ assigns types to events.
- $\text{eatype} : \mathcal{E} \to \mathcal{U}_{etype}$ assigns event types to event attributes.
- $\text{oatype} : \mathcal{O} \to \mathcal{U}_{otype}$ assigns object types to object attributes.
- $\mathcal{O2O}$ represents qualified object-to-object relations.

In this paper, we consider OCEL 2.0 logs represented in the JSON format.

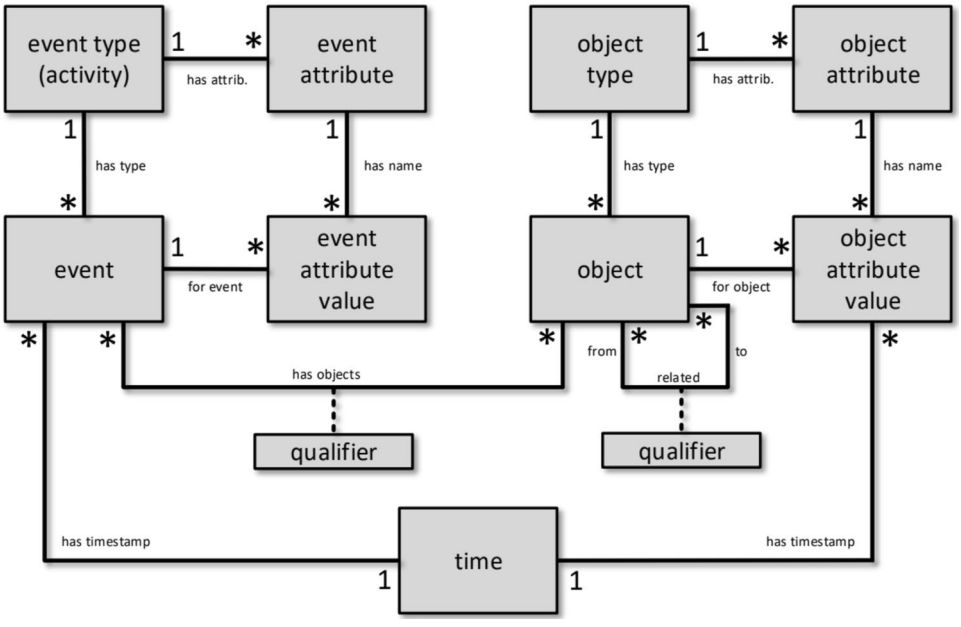


**Fig. 1** OCEL 2.0 metamodel (Berti et al. 2024a)

### Large language models

LLMs are computational models that have significantly transformed NLP using deep neural networks for text understanding and generation (Naveed et al. 2025). Built on advanced *transformer*-based architectures and trained on extensive datasets, LLMs leverage self-attention mechanisms to learn language patterns to predict and generate coherent text by estimating token likelihood within a broader context (Vaswani et al. 2017). Notably, researchers have found that increasing the model's *size* (i.e., the number of parameters) enhances performance and introduces unique capabilities, such as *in-context learning*, *instruction following*, and *step-by-step reasoning* (Naveed et al. 2025). The evolution of LLMs marked substantial progress in the NLP field, with models such as GPT (OpenAI 2023), LLaMA (Dubey et al. 2024), Mistral (Jiang et al. 2023), and Gemma (Rivière et al. 2024) excelling across various tasks from natural language question-answering (Kojima et al. 2022) to code explanation and generation (Chirkova et al. 2021).

Recent research focused on enhancing architectures, training methodologies, fine-tuning techniques, and model scalability (Naveed et al. 2025). Yet, challenges persist in ensuring accurate and contextually relevant responses from LLMs due to potential biases and outdated knowledge inherited from the training data (Gao et al. 2023; McKenzie et al. 2023) and hallucinations (Huang et al. 2025).

#### *Retrieval augmented generation*

A prominent approach to enhance the accuracy and credibility of LLMs without the complexity of fine-tuning is *Retrieval-Augmented Generation* (RAG) (Lewis et al. 2020), combining information retrieval with in-context learning (Dong et al. 2024). With RAG, the user's query triggers the retrieval of relevant information from external sources via search algorithms, which is combined into the prompt to provide additional context for the LLM (Gao et al. 2023).

More formally, RAG involves a *retriever* and a *generator*, operating through three steps: *retrieval*, *augmentation*, and *generation*. In the retrieval phase, the query $x$ from the user is utilized for retrieving relevant context (text documents $z$) from an external knowledge source using a retriever $p_\eta(z|x)$, where parameters $\eta$ return distributions over the text documents given $x$. Subsequently, the question is embedded into a vector space and stored in a vector database. In the augmentation stage, the $k$ most relevant documents are retrieved from the database based on vector similarity and merged with the initial query into a prompt template. Eventually, the prompt is provided to the LLM to generate responses based on the retrieved context.

To better understand the RAG retrieval stage, we need to introduce *embedding models*.

#### *Embedding models*

An *embedding model* is a machine learning model that converts discrete, categorical data (e.g., text, images, or, in this work, trace prefixes) into continuous vectors, typically in a high-dimensional space (Wang et al. 2019). These models aim to capture the semantic relationships and underlying structure of the data.

Early approaches, such as traditional TF-IDF and static Word2Vec (Mikolov et al. 2013), focused on generating embeddings for individual words. However, modern, dynamic, and contextualized approaches leverage advanced architectures to create

embeddings for entire sentences and paragraphs. Models based on the transformer (Vaswani et al. 2017), such as BERT (Koroteev 2021), use self-attention mechanisms to weigh the importance of different words in a sequence, capturing nuanced semantic relationships.

Sentence embedding models are optimized for tasks like semantic search, clustering, and paraphrase identification. For example, models from the *Sentence-Transformers*[2] library are widely used due to their efficiency and strong general-purpose performance. More recently, novel long-context embedding models have emerged for encoding longer sequences of tokens at a higher inference cost (Günther et al. 2025).

Embedding models form the backbone of the RAG pipelines (Gao et al. 2023): by converting both the user's query and the documents in a knowledge base into embeddings, the pipeline can efficiently find the most relevant information using *vector similarity search*. In our work, this retrieved context, i.e., relevant statistics about the object-centric event log, is then used by an LLM to generate a factual and grounded response.

## Related work

In recent years, the integration of conversational techniques in process mining for event log analysis has gained considerable attention, holding the potential to make process mining outcomes more accessible to general users (Casciani et al. 2024). While several works have explored natural language interfaces for process mining, they have primarily focused on traditional event logs in the XES format or on extracting process information from textual documentation.

For instance, (Yeo et al. 2022) proposes a conversational interface enabling data extraction in healthcare, bypassing the need for a query language. Similarly, (Kobeissi et al. 2023) employs graph-based storage and queries using the Cypher language to improve natural language querying experiences for non-technical analysts. Additionally, (Barbieri et al. 2023) develops a taxonomy of process mining questions and introduces a rule-based querying interface for XES event logs.

Beyond natural language querying, prior research has also focused on process discovery and the interpretability of process mining outcomes. (Han et al. 2020) presents a method for automatically discovering process models from textual documentation using a neural network with the Ordered Neurons LSTM architecture. The work in (Fahland et al. 2023) improves process discovery from event logs with causality analysis and explainable Artificial Intelligence (AI), leveraging LLMs to improve the interpretability of business process outcomes. (Rebmann and van der Aa 2021) introduces a technique for automatically extracting process information, including resources and objects, through semantic role labeling and attribute classification. Moreover, (Resinas et al. 2023) combines NLP techniques with tailored matching strategies to integrate textual descriptions with event logs and define measurable Process Performance Indicators (PPIs).

More recent studies have increasingly focused on the role of LLMs in Business Process Management (BPM) and process mining tasks. (Grohs et al. 2023) demonstrated that LLMs can address multiple BPM tasks, including the extraction of imperative and declarative models from textual descriptions, while (Estrada-Torres et al. 2024) provided a comprehensive mapping of how LLMs are being applied within the BPM lifecycle. The

[2] Sentence-Transformers Documentation: https://www.sbert.net/

benchmark proposed by (Berti et al. 2024b) further evaluated the performance of open-source and commercial LLMs on process mining tasks, highlighting both their promise and current limitations, whereas (Rebmann et al. 2025) explored their capability to tackle semantics-aware tasks such as process discovery and anomaly detection. In parallel, (Pasquadibisceglie et al. 2024) introduced a predictive process monitoring approach that reformulates event logs into textual narratives for suffix prediction, leveraging fine-tuned LLMs with explainability features.

The introduction of LLM-based natural language interfaces has also progressed. For example, (Barbieri et al. 2024) proposed a hybrid framework combining LLMs with traditional query-based techniques to support scalable and precise question-answering over event logs. Instead, (Agostinelli et al. 2025) presented PPIPilot, an LLM-based system for automatically suggesting and computing measurable PPIs from event logs, demonstrating how LLMs can directly contribute to decision-support tasks in organizations.

Along with the advancements in conversational, explainable, and predictive process mining, a distinct line of research focused on process query languages aimed at extracting insights from event logs. For instance, (Vogelgesang et al. 2022) introduces the Process Querying Language (PQL), an SQL-based approach that integrates with the data model, avoiding the need for explicit table joins. Beyond flat event data, some approaches have been developed to address object-centric event logs. For example, (Esser and Fahland 2021) proposes storing multi-dimensional event data in graph databases and utilizing Cypher to retrieve entities and subgraphs. Similarly, (Schönig et al. 2016) employs SQL-based techniques to identify DECLARE constraints within event logs. However, extending these methods to accommodate more complex constructs significantly increases execution times, underscoring the performance limitations in querying multi-dimensional event data (Schönig et al. 2016; Riva et al. 2023). More recently, research has also investigated structural challenges in object-centric process mining. In particular, (Liss and van der Aalst 2025) proposed a multilevel resource detection method to clarify process areas and hierarchies in object-centric logs, improving the interpretability of mined models.

Taken together, these studies demonstrate a growing trend in leveraging LLMs and natural language interfaces to make process mining more accessible and interpretable. However, while substantial progress has been made, none of the above contributions directly target the conversational analysis of object-centric event logs. Our framework addresses this gap by developing a framework for implementing conversational interfaces for OCEL 2.0 event logs, enhancing accessibility for non-technical users while maintaining the semantic richness of object-centric data.

## Methodology

In this section, we describe the RAG-based framework for querying information from the object-centric event log by extending the technique proposed in (Bernardi et al. 2024) to consider business process execution aspects. As illustrated in Fig. 2, the framework is structured in two stages, namely the *Event Log Preprocessing* and the *Question Answering*.

The *Event Log Preprocessing* stage is an offline step that prepares the data for the subsequent retrieval. It starts with the preprocessing module, which extracts statistics about the overall process execution, events, objects, and their relationships from the raw

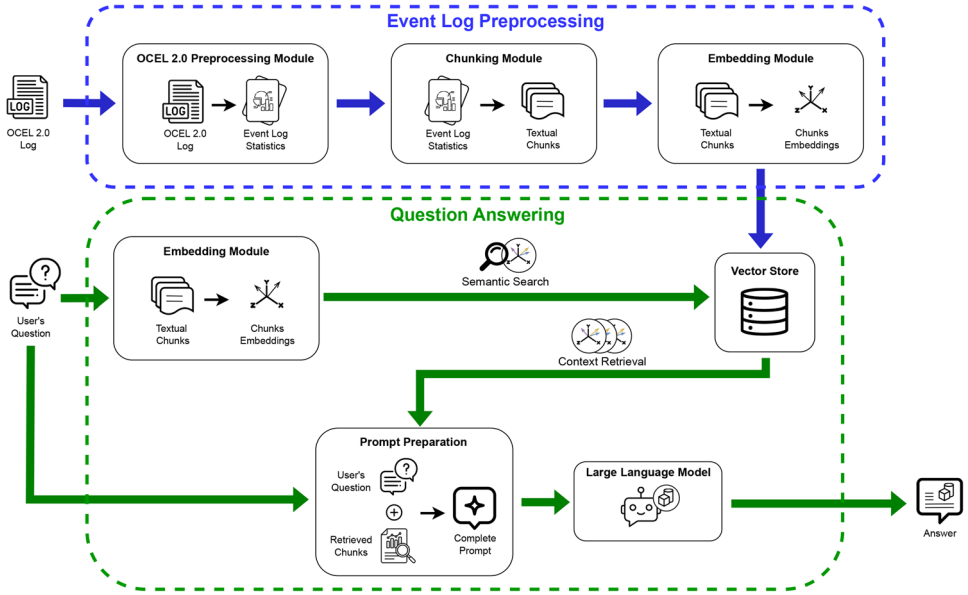


**Fig. 2** The overall architecture of the framework, consisting of the *event log preprocessing* (top, blue, dashed box) and the *question answering* (bottom, green, dashed box) stages. The question from the user is used to retrieve relevant information derived from the preprocessing of the event log. Such chunks are subsequently added to the prompt for the LLM through RAG to support it in providing a faithful answer

OCEL 2.0 log file. The chunking module then segments these facts and statistics into more manageable yet semantically coherent textual chunks, which are converted into numerical embeddings and stored in a vector database.

The *Question Answering* stage is the online component interacting with the human. When a user submits a question in natural language, the latter is transformed into a vector embedding. A semantic search is then performed to retrieve the most relevant chunks from the vector store, which are combined with the user's query into a single augmented prompt. This prompt is passed to the LLM, which generates a context-grounded answer.

The framework was implemented leveraging well-known Python libraries such as *Langchain*[3] for the LLM architecture, and *pm4py*[4] and *pandas*[5] for the object-centric processing. The following subsections provide an in-depth discussion of each phase.

### Event log Preprocessing

The preprocessing stage extracts from an OCEL 2.0 event log $\mathcal{L}$ in JSON format a structured collection of statistics regarding the process execution, following the definitions given in Sect. 2.2. Each category has its own internal representation:

- *Aggregate global statistics about the business process execution.* This heterogeneous category includes the total duration of the overall execution, the earliest and latest recorded timestamps, and the number of events, activities, object types, and individual objects. Additionally, the occurrences of each activity and object type are computed. Example of

[3] Langchain website: https://www.langchain.com/

[4] PM4Py documentation: https://processintelligence.solutions/static/api/2.7.11/index.html

[5] Pandas documentation: https://pandas.pydata.org/docs/

elements are: the *total duration 944 days 11 hours*, the *activities occurrences* $\{create\_goods\_receipt : 4042, create\_invoice\_receipt : 1941, ..., approve\_purchase\_requisition : 607\}$, etc.

- *Event statistics*. For each event in the log, the preprocessing module determines the manipulated object types and their number of instances. This group of statistics is in the form of a dictionary, e.g., $\{event_1 : \{material : 4, purchase\_requisition : 1\}, event_{100} : \{quotation : 1, purchase\_order : 1, ...\}, ...\}$.
- *Object statistics*. Both information about the object types and the instantiated objects are derived from the log. Concerning individual objects, this includes the activities that manipulate them, their lifecycle start and end timestamps, corresponding duration, and interactions with other objects during their lifecycle. From the object type perspective, the list of objects and the activities that operate on them are identified and stored. Furthermore, this step computes statistics about the values of objects' attributes for each object type on the flattened version of the event log. For the attributes of type date, the minimum and maximum dates registered, along with the event and individual object generating them, are recorded. For numerical attributes, this preprocessing extracts their minimum, maximum, and average values, with the individual object and event reporting them. For textual attributes, the list of possible values observed during execution is extracted. Moreover, for each object type, the resources manipulating them and their working calendar are discovered on the flattened versions of $\mathcal{L}$. For instance, samples from this collection are structured as tuples $t = \langle ocel\_oid, activities, start, end, duration \rangle$, e.g., $\langle goods\_receipt_0, [create\_goods\_receipt, create\_invoice\_receipt, ...]$, 2022/04/11_08:52, 2022/04/18_10:05, 609180$\rangle$.
- *Timestamp statistics*. For each recorded timestamp, we extract and store in this collection the activities executed and the objects manipulated. Elements here are tuples in the form of $t = \langle timestamp, activity, ocel\_oid \rangle$, e.g., $\langle$ 2022/04/01_09:26, $[create\_purchase\_requisition, ...], [material_0, purchase\_requisition_0]\rangle$.

Afterward, the extracted statistics are segmented into manageable, semantically coherent chunks. The segmentation strictly depends on the internal structure of each category in the above collection of statistics. Each chunk represents a single element inside a category, according to the above definitions (e.g., $\langle goods\_receipt_0, [create\_goods\_receipt, create\_invoice\_receipt, ...]$, 2022/04/11 08 : 52, 2022/04/18 10 : 05, 609180$\rangle$ for the object statistics, and so on). This approach preserves the integrity and meaningfulness of the textual chunks, enabling the retrieval of specific information in the subsequent Question Answering step in a granular manner, as reported in the "*Retrieved Context*" of Fig. 3.

Each chunk is then transformed into an embedding using the *all-MiniLM-L12-v2*[6] model from the *sentence-transformers* library. It maps each chunk into a 384-dimensional dense vector space, capturing semantic similarity across heterogeneous data. The choice of this model was motivated by its balance of accuracy, inference speed, and low memory usage, making it suitable even for scenarios with limited computational resources or latency constraints.

[6] all-MiniLM-L12-v2 on HuggingFace: https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

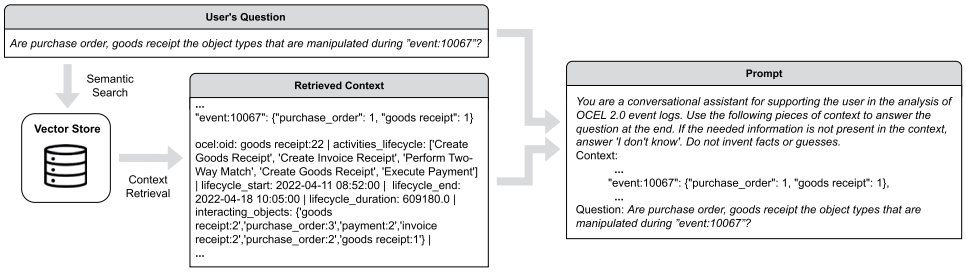


**Fig. 3** The prompt's composition. The user is asking whether objects of type *"purchase order"* and *"goods receipt"* are manipulated during the *"event:10067"* in the log. The framework retrieves from the vector index chunks detailing the object types involved in *"event:10067"* and providing lifecycle details for a *"goods receipt:22"*, and the augmented prompt is then sent to the LLM

The resulting embeddings are then stored in a vector database (in our case, *Qdrant*[7]), enabling efficient retrieval during query answering. The embeddings are accompanied by a metadata field that uniquely identifies the type of chunks they refer to (e.g., whether the stored information pertains to a specific object, event, timestamp, and so on). This improves the precision of the retrieval of specific information within the statistics collection extracted from the object-centric log $\mathcal{L}$ by combining the semantically retrieved context with the results of metadata search as needed, based on the user's query.

### Question answering

The subsequent stage concerns receiving, processing, and answering the users' questions related to the event log. In particular, the framework utilizes semantic search based on *cosine similarity* to retrieve the top-$k$ contextually relevant chunks from the vector database based on the query, for assisting the LLM in answering accurately while respecting its context window. Our empirical results indicate that retrieving the top three to five chunks based on semantic similarity to the query is sufficient to generate a satisfactory answer. This design choice ensures that the prompt contains only highly relevant information without requiring large context windows, making the framework compatible even with smaller LLMs and keeping inference times low.

The retrieved chunks, along with the initial query, are combined into a prompt, as shown in Fig. 3. This prompt is refined by adding a system message to instruct the LLM on how to respond, avoiding making up answers when sufficient contextual information is unavailable.

Such a prompt is composed by unifying the question with the retrieved context. In the example reported in Fig. 3, the user is asking whether objects of type *"purchase order"* and *"goods receipt"* are manipulated during the "$event_{10067}$" in the log. The framework performs a semantic search on the index in the vector store using the embedding vector of the question to retrieve relevant data chunks (e.g., one directly stating the object types involved in "$event_{10067}$" and another providing lifecycle details for a "$goods\_receipt_{22}$" object).

Then, the prompt is refined, adding a *system message* to instruct it on how to answer faithfully to the question, i.e., by using the retrieved context and avoiding answering in the absence of relevant information.

[7] Qdrant is a well-known open-source vector similarity search engine and database. Website: https://qdrant.tech/qdrant-vector-database/

Ultimately, the prompt is fed into the LLM, which leverages the provided context to generate responses that are contextually coherent with the initial question and grounded in the specific details of the object-centric event log under analysis.

To illustrate how the instructions to the LLM influence its behavior, we report the three main interaction scenarios. The first shows a case where the retrieved statistics support the query and allow the model to return a positive answer.

**Information Found - Positive Answer**

**System Message:** You are a conversational assistant for supporting the user in the analysis of OCEL 2.0 event logs. Use the following pieces of context to answer the question at the end. If the needed information is not present in the context, answer *"I don't know"*. Do not invent facts or guesses.

**Context:**
1. (ocel:oid: goods receipt:1004, activities_lifecycle: ["Create Goods Receipt", "Create Invoice Receipt", "Perform Two-Way Match", "Execute Payment", "Execute Payment", "Create Goods Receipt"], lifecycle_start: 2023-08-25 15:55, lifecycle_end: 2023-09-03 01:06, lifecycle_duration: 724260.0, interacting_objects: "purchase order:831", "payment:489", "goods receipt:1006", "goods receipt:1005", "invoice receipt:1004", "purchase order:832") **...**

**Question:** *What are the activities involved in the lifecycle of ocel:oid "goods receipt:1004"?*

**Answer:** The lifecycle activities are: ['Create Goods Receipt', 'Create Invoice Receipt', 'Perform Two-Way Match', 'Execute Payment', 'Execute Payment', 'Create Goods Receipt'].

The second interaction shows a scenario where the retrieved statistics contradict the query, leading to a negative answer.

**Information Found - Negative Answer**

**System Message:** You are a conversational assistant for supporting the user in the analysis of OCEL 2.0 event logs. Use the following pieces of context to answer the question at the end. If the needed information is not present in the context, answer *"I don't know"*. Do not invent facts or guesses.

**Context:**
1. (ocel:oid: goods receipt:1004, activities_lifecycle: ["Create Goods Receipt", "Create Invoice Receipt", "Perform Two-Way Match", "Execute Payment", "Execute Payment", "Create Goods Receipt"], lifecycle_start: 2023-08-25 15:55, lifecycle_end: 2023-09-03 01:06, lifecycle_duration: 724260.0, interacting_objects: "purchase order:831", "payment:489", "goods receipt:1006", "goods receipt:1005", "invoice receipt:1004", "purchase order:832") **...**

**Question:** *Is the activity Create Purchase Requisition in the lifecycle of ocel:oid "goods receipt:1004"?*

**Answer:** No. The lifecycle activities for goods receipt:1004 do not include "Create Purchase Requisition".

The third example demonstrates a situation where the framework cannot provide an answer because the retrieved context does not contain the required information. This

may occur either because no relevant chunks are retrieved or because the retrieved chunks, although present, are semantically unrelated to the question due to errors in the similarity-based retrieval. In both situations, the framework acknowledges the absence of supporting evidence rather than producing an unsupported answer.

**Information Not Found or Not Relevant - No Answer**

**System Message:** You are a conversational assistant for supporting the user in the analysis of OCEL 2.0 event logs. Use the following pieces of context to answer the question at the end. If the needed information is not present in the context, answer *"I don't know"*. Do not invent facts or guesses.

**Context:** No relevant context found for the given query.

**Question:** *What are the activities involved in the lifecycle of ocel:oid "goods receipt:42022"?*

**Answer:** I don't know.

As we can see, the framework properly answers if the information (whether confirmed or not by the retrieved context) is present and does not respond if it found no evidence in the context (e.g., the object "$goods\_receipt_{42022}$" does not exist in the log).

## Experimental evaluation

In this section, we discuss the experiments and the results obtained in answering the RQs introduced in Sect. 1. For **RQ1**, we aim to determine the effectiveness of an LLM-based interface in providing precise answers about the object-centric event log from a quantitative standpoint. Conversely, for **RQ2**, we performed a qualitative evaluation of the framework with human users to get feedback about the completeness and usefulness of its answers about the OCEL 2.0 event log.

### Event log

The event log selected for the evaluation is one of the reference OCEL 2.0 logs[8]. It describes the execution of a *Procure-To-Pay* (P2P) business process, starting from a purchase request up to the execution of the payment. This simulated event log offers a realistic representation of a P2P process by relying on real SAP transactions and object types. Figure 4 reports an extract of the log, flattened on the *"invoice receipt"* object type.

### Dataset

For the quantitative evaluation, we leveraged the P2P event log to develop a dataset $\mathcal{D} = \{(q_1, y_1), (q_2, y_2), \ldots, (q_N, y_N)\}$ that comprises $N = 2552$ pairs of questions and answers, categorized into the four major execution perspectives captured by the object-centric log. These questions reflect the structure previously outlined in Sect. 4 for the information extracted during the preprocessing step, i.e., knowledge about *objects*, *events*, *timestamps*, and *global statistics* of $\mathcal{L}$). For each row, the first value represents the question concerning the log, while the second value is a boolean indicating the expected response (see Fig. 4): this allows us to objectively assess if the framework answers correctly in the quantitative evaluation.

[8] The Procure-to-Pay log is available at: https://www.ocel-standard.org/event-logs/simulations/p2p/

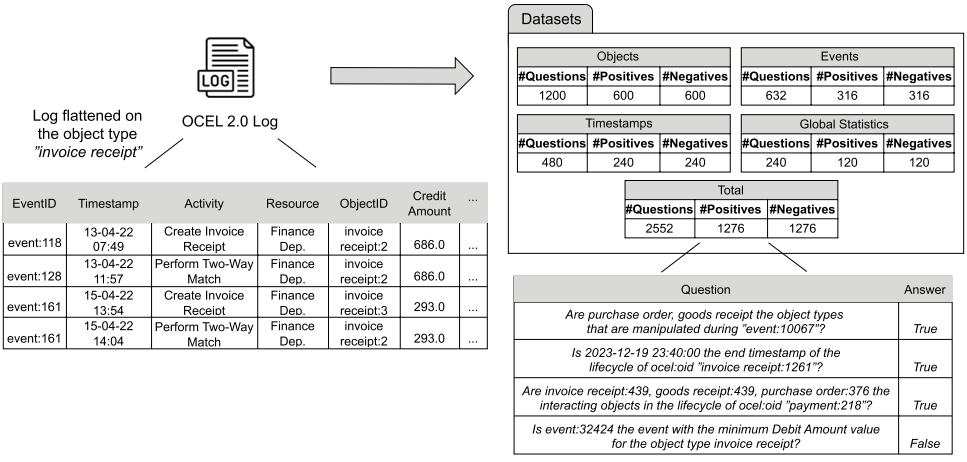


**Fig. 4** Generation of the evaluation dataset from the OCEL 2.0 log. From the event log, a dataset consisting of question-answer pairs is derived. It is divided equally between positive (true) and negative (false) examples. The bottom-right table provides examples of binary questions in the dataset

We generated $\mathcal{D}$ by randomly sampling a specified number of objects, events, and timestamps from the OCEL 2.0 log in JSON to populate predefined template questions for each category, with *True* as the answer. Then, we computed global statistics about the log (e.g., total process duration, object with the highest credit amount, etc.) to complete the positive questions. Finally, an equal number of negative questions (i.e., with a *False* answer) are generated by randomly altering some details in the positive questions. These pairs are categorized into the mentioned four distinct perspectives of object-centric analysis and randomly shuffled within each dataset.

We now exemplify two samples (one positive and one negative) from the dataset to better clarify their composition. For instance, the query *"Are the purchase order and goods receipt object types manipulated duringevent*$_{10067}$?" has a *True* answer, since in the log instances of both object types are indeed manipulated in that specific event. Conversely, the query *"Isevent*$_{32424}$*the event with the minimum debit amount value for the object type invoice receipt?"* foresees a *False* answer, as another event in the log manipulates an invoice receipt object with a smaller debit amount than the one observed in *event*$_{32424}$.

We provide the dataset in two variants: a comprehensive dataset containing all the questions randomly shuffled, and separate files reflecting the different types of questions (i.e., global statistics, events, objects, and timestamps information)[9].

### Experimental settings

Guided by the previously introduced RQs, we evaluated the framework covering qualitative and quantitative aspects and leveraging multiple performance metrics.

The framework was examined in its ability to assist the analysis of the object-centric event log from various perspectives through the dataset $\mathcal{D}$, focusing on global information and knowledge related to the objects, events, and timestamps within the log. To ensure a comprehensive assessment, we selected a diverse set of LLMs, comparing the performance of open-source models against their closed-source counterparts in this type of analysis. In particular, we considered models from the Meta Llama (Dubey et al.

[9] The datasets are available at: https://doi.org/10.5281/zenodo.18086659

2024) and Mistral (Jiang et al. 2023) families, a Qwen model (Yang et al. 2024), a Gemma model from Google (Rivière et al. 2024), and a variant of GPT-4 from OpenAI (OpenAI 2023).

The quantitative evaluation used the questions drawn from the dataset and was designed as a *binary classification* problem (Géron 2022) to enable a rigorous assessment of the answers provided by the framework. We define the vector database as $\mathcal{C} = \{c_1, \ldots, c_M\}$, containing the $M$ contextual chunks of the statistics from the event log $\mathcal{L}$. Given the above dataset $\mathcal{D}$ and the vector database $\mathcal{C}$, the framework is fed with the prompt $p_i = (s, \hat{C}, q_i)$ for each question $q_i \in \mathcal{D}$. In particular, $s$ is a system message to instruct the LLM to respond in a binary way $\hat{y}_i = 1$ (i.e, *true*) or $\hat{y}_i = 0$ (i.e., *false*) based on the context chunks $\hat{C} = \{c_j, \ldots, c_k\} \subset C,\ j, k \in \{1, \ldots, M\}$ semantically retrieved from the vector database $\mathcal{C}$. The framework's answers that match the expected positive responses ($\hat{y}_i = y_i = 1$) are labeled as *true positives* (*TP*), and those that match the expected negative answers ($\hat{y}_i = y_i = 0$) are labeled as *true negatives* (*TN*). In contrast, *false positives* (*FP*) occur when the framework produces positive answers when negative ones are expected ($\hat{y}_i = 1 \wedge y_i = 0$), and *false negatives* (*FN*) appear when the framework generates negative responses despite positive expectations ($\hat{y}_i = 0 \wedge y_i = 1$).

Assuming these definitions, we employed the metrics defined in Table 1 to provide a multi-perspective and rigorous evaluation of the framework's performance. While *accuracy* offers a straightforward measure of the overall percentage of correct answers, it can be insufficient on its own. Therefore, we include *precision* and *recall* to understand the nature of the framework's errors: precision measures the reliability of the positive predictions; a high precision means users can trust the facts the framework confirms. Conversely, recall measures the ability to identify all true facts. The *F1-score* provides their harmonic mean, offering a single, balanced measure of a model's ability to be both correct and comprehensive.

To further strengthen the evaluation, we also considered standard classification metrics like the *Matthews correlation coefficient* (MCC) and the *Area Under the ROC Curve* (AUC). MCC is a robust metric producing a high score only if the framework obtains good results in all four categories of the confusion matrix (*TP*, *TN*, *FP*, *FN*). Instead, the AUC provides a measure of the model's ability to distinguish between true and false statements across all decision thresholds, making it an indicator of the framework's underlying discriminative power, independent of any bias towards predicting a specific class.

**Table 1** Evaluation metrics and their formulas

| Metric | Formula | Definition |
|---|---|---|
| *Accuracy* | $\frac{TP+TN}{TP+FP+TN+FN}$ | Ratio of correctly predicted instances to the total number of instances. |
| *Precision* | $\frac{TP}{TP+FP}$ | Ratio of correctly predicted positive instances to the total predicted positives. |
| *Recall* | $\frac{TP}{TP+FN}$ | Ratio of correctly predicted positive instances to all real positives. |
| *F1-score* | $\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$ | Harmonic mean of precision and recall. |
| *MCC* | $\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$ | Correlation coefficient between the real and predicted classifications. |
| *AUC* | $\int_0^1 \frac{TP}{TP+FN}\, d\left(\frac{FP}{FP+TN}\right)$ | Area under the ROC curve that plots the trade-off between the true positive rate and the false positive rate across different classification thresholds. |

We addressed **RQ1** by conducting a structured *quantitative* evaluation of the framework's ability to answer inquiries about the P2P event log $\mathcal{L}$ across several object-centric perspectives. We systematically assessed two main factors: the *choice of LLM* and the *prompt engineering strategy* employed in the framework. Thus, we performed an ablation study on the prompt[10], following well-established prompt engineering conventions (Sahoo et al. 2024) and evaluating a set of LLMs across four distinct prompting scenarios:

- The *zero-shot* prompt served as our baseline, providing the LLM with only the user's question and the retrieved context. It included no examples or the instructional guardrail for factual responses detailed in Sect. 4.2.
- The *few-shot* prompt augmented the baseline by incorporating examples of how the retrieved information can be used to answer questions about the log, one for each considered object-centric perspective.
- The *zero-shot with guardrails* prompt added the instruction to enforce a factual response to the baseline zero-shot prompt.
- The *few-shot with guardrails* scenario combined both enhancements, providing the LLM with examples and factual guardrails.

For the assessment, we used a *binary oracle* for computing the selected classification metrics by comparing each response against a ground-truth answer. To ensure robustness, all results were averaged over *five independent runs* under each experimental setting and the results were averaged. We fixed the number of retrieved chunks to three, providing the best balance between completeness of information and computational efficiency.

Using the best configuration of LLM (i.e., *Ministral-8B*) and prompt (i.e., *few-shot with guardrails*) identified through the quantitative evaluation (see Sect. 5.4), we tackled **RQ2** by carrying out a qualitative study of the framework following the guidelines outlined in van der van der Lee C et al. (2021). Our goal was to gain insights into the answers provided by the framework to more subtle queries in natural language about facts and statistics from the OCEL 2.0 log. This simulates the interaction with potential users in a real workplace and is instrumental for understanding their perception of its strengths and weaknesses, and for identifying areas for improvement.

Therefore, we developed a *survey*. The survey was distributed to 12 participants from Sapienza University of Rome, including 2 Master's students and 10 academics (7 doctoral students and 3 professors), all with a background in Computer Science or Management Engineering. Before interacting with the framework, participants were provided with background information on process mining, traditional event logs, the limitations of conventional techniques, and the advantages of object-centric analysis in natural language. As a subset of the sample was not completely familiar with object-centric concepts, we also presented an overview of the kinds of queries that can be formulated over the P2P event log introduced in Sect. 5.1. Specifically, we illustrated a representative example question for each statistics category extracted during preprocessing, to ensure that participants understood the type of information available in the log.

Participants were then invited to freely experiment with the framework by asking any natural language questions about the log they wished, at least three. No predefined tasks

[10] Complete prompts are available at https://github.com/angelo-casciani/Conversational-OCEL2

or objectives were provided. Consequently, the number and diversity of questions varied across participants. The interactions with the framework were logged and later analyzed.

After the interaction phase, participants were asked to answer three open-ended questions: (*i*) to comment on the *clarity*, *usefulness*, and *alignment* of the answers with their original queries (van der van der Lee C et al. 2021); (*ii*) to provide feedback on the *strengths* and *weaknesses* of the framework; and (*iii*) to suggest potential *areas for improvement*. Responses were collected anonymously. We then analyzed the questions posed by the users, the answers generated by the framework, and the participants' written feedback.

The experiments were performed using a workstation running the Linux/Ubuntu 22.04.3 LTS operating system and equipped with a NVIDIA A100 GPU.

### Evaluation results

We now examine the outcomes of the tests conducted employing the framework under the previously defined experimental settings to address the RQs.

#### *Quantitative evaluation*

Table 2 reports the evaluation results of the framework embedding various LLMs and prompting strategies, showing how its performance in providing faithful answers about the event log $\mathcal{L}$ is influenced by these design choices.

A first observation that can be made is that the prompting strategy is as crucial as the choice of LLM itself. The results show that, overall, the *few-shot with guardrails* prompt tends to improve performance for the considered LLMs among the four prompting scenarios. While the baseline zero-shot prompts offer a reasonable starting point for capable models like *Mistral-7B-v0.3*, the introduction of guardrails provides an important performance enhancement for the majority of models, ensuring they adhere to the task's constraints, such as providing a boolean answer based on contextual facts. This is most evident with *Mistral-7B-v0.2*, whose accuracy considerably increases from 0.54 in the *zero-shot* scenario to 0.96 with the addition of guardrails. The inclusion of relevant task examples with the few-shot prompts further improves the performance of top models, pushing them toward near-perfect scores.

The choice of LLM also remains an important factor. In the optimal *few-shot with guardrails* prompting scenario, *Ministral-8B* stands as the best model, achieving the highest scores in nearly every metric, including the highest accuracy (i.e, 0.98) and MCC (i.e., 0.96). Following closely, there is a group of competitive models, including *Llama-3-8B*, *Qwen2.5-7B*, and *gpt-4o-mini*, all of which demonstrate reliable performance with MCC scores above 0.93.

Conversely, the evaluation confirms the expected correlation between model size and capability: the smaller model in terms of parameter size among the considered ones, i.e., *Llama-3.2-1B*, always yields the lowest performance across all scenarios. This result indicates that the *inverse scaling* phenomenon, wherein a smaller model outperforms a larger one on a particular task, is not observed for this dataset (McKenzie et al. 2023). Indeed, smaller models may have limited capacity to interpret the retrieved object-centric chunks, sometimes focusing on irrelevant information or failing to connect the question semantics with the retrieved facts. They may also struggle to correctly map the conclusions derived from the context to the required boolean answer in our evaluation

**Table 2** Framework evaluation with various LLMs on the *complete* dataset $\mathcal{D}$ across four different prompt engineering scenarios. For each metric within each scenario, the best results are highlighted in bold, while the runner-ups are underlined

| **LLM** | **Zero-Shot** | | | | | | **Zero-Shot with Guardrails** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** |
| *Llama-3-8B* | 0.92 | 0.95 | 0.89 | 0.92 | 0.84 | 0.92 | 0.96 | 0.95 | 0.97 | 0.96 | 0.92 | 0.96 |
| *Llama-3.1-8B* | 0.87 | 0.96 | 0.77 | 0.86 | 0.75 | 0.87 | **0.97** | 0.96 | **0.98** | **0.97** | **0.94** | **0.97** |
| *Llama-3.2-1B* | 0.51 | 0.59 | 0.08 | 0.14 | 0.05 | 0.51 | 0.58 | 0.67 | 0.33 | 0.44 | 0.19 | 0.58 |
| *Llama-3.2-3B* | 0.84 | 0.86 | 0.83 | 0.84 | 0.69 | 0.84 | 0.91 | 0.89 | 0.93 | 0.91 | 0.82 | 0.91 |
| *Mistral-7B-v0.2* | 0.54 | **1.00** | 0.07 | 0.13 | 0.19 | 0.53 | 0.96 | 0.96 | 0.95 | 0.96 | 0.91 | 0.96 |
| *Mistral-7B-v0.3* | **0.97** | 0.95 | 0.98 | **0.97** | **0.94** | **0.97** | 0.96 | 0.96 | 0.97 | 0.96 | 0.93 | 0.96 |
| *Mistral-Nemo* | 0.89 | 0.98 | 0.80 | 0.88 | 0.80 | 0.89 | 0.91 | 0.96 | 0.86 | 0.91 | 0.83 | 0.91 |
| *Ministral-8B* | 0.96 | 0.94 | **0.99** | 0.96 | 0.93 | 0.96 | 0.91 | 0.97 | 0.86 | 0.91 | 0.83 | 0.91 |
| *Qwen2.5-7B* | 0.95 | 0.98 | 0.92 | 0.95 | 0.90 | 0.95 | **0.97** | 0.97 | 0.97 | **0.97** | 0.93 | **0.97** |
| *gemma-2-9b* | 0.72 | 0.92 | 0.48 | 0.63 | 0.50 | 0.72 | 0.71 | **1.00** | 0.39 | 0.56 | 0.49 | 0.69 |
| *gpt-4o-mini* | 0.96 | 0.96 | 0.96 | 0.96 | 0.92 | 0.96 | 0.96 | 0.97 | 0.96 | 0.96 | 0.92 | 0.96 |
| | **Few-Shot** | | | | | | **Few-Shot with Guardrails** | | | | | |
| **LLM** | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* |
| *Llama-3-8B* | 0.96 | 0.96 | 0.95 | 0.96 | 0.92 | 0.96 | 0.97 | 0.95 | **0.99** | 0.97 | 0.94 | 0.97 |
| *Llama-3.1-8B* | 0.93 | 0.96 | 0.90 | 0.93 | 0.86 | 0.93 | 0.96 | 0.96 | 0.95 | 0.96 | 0.92 | 0.96 |
| *Llama-3.2-1B* | 0.51 | 0.55 | 0.13 | 0.21 | 0.04 | 0.51 | 0.72 | 0.78 | 0.63 | 0.69 | 0.46 | 0.72 |
| *Llama-3.2-3B* | 0.92 | 0.94 | 0.91 | 0.92 | 0.85 | 0.92 | 0.91 | 0.94 | 0.88 | 0.91 | 0.83 | 0.91 |
| *Mistral-7B-v0.2* | 0.64 | 0.93 | 0.31 | 0.46 | 0.39 | 0.64 | 0.80 | 0.94 | 0.64 | 0.76 | 0.63 | 0.80 |
| *Mistral-7B-v0.3* | 0.96 | 0.95 | 0.97 | 0.96 | 0.93 | 0.96 | 0.94 | 0.91 | 0.98 | 0.94 | 0.89 | 0.94 |
| *Mistral-Nemo* | 0.87 | **0.99** | 0.74 | 0.85 | 0.77 | 0.87 | 0.92 | **0.99** | 0.85 | 0.91 | 0.85 | 0.92 |
| *Ministral-8B* | **0.97** | 0.95 | 0.98 | **0.97** | **0.94** | **0.97** | **0.98** | **0.99** | 0.97 | **0.98** | **0.96** | **0.98** |
| *Qwen2.5-7B* | 0.90 | 0.96 | 0.84 | 0.90 | 0.82 | 0.90 | 0.97 | 0.96 | 0.97 | 0.97 | 0.94 | 0.97 |
| *gemma-2-9b* | 0.55 | 0.53 | **0.99** | 0.69 | 0.22 | 0.55 | 0.80 | 0.88 | 0.62 | 0.74 | 0.65 | 0.80 |
| *gpt-4o-mini* | 0.96 | 0.97 | 0.95 | 0.96 | 0.92 | 0.96 | 0.97 | 0.98 | 0.95 | 0.97 | 0.93 | 0.97 |

setup, and tend to be more sensitive to distractors within the retrieved text, even when only three chunks are provided.

We further investigated **RQ1** by examining the effects of specific aspects of the object-centric event log on the reliability of the answers. The outcomes of this evaluation are divided according to the considered object-centric perspective of the dataset, i.e., *global statistics*, *events*, *objects*, and *timestamps* information related to the P2P event log.

Table 3 shows the evaluation results in answering questions related to *global information* about the log $\mathcal{L}$. The generally lower scores across all models compared to the complete dataset underscore the increased difficulty of this perspective.

Regarding the prompting strategy, the baseline *zero-shot* scenario often led models to show low recall. For instance, *Llama-3-8B* and *Mistral-7B-v0.2* achieved perfect precision but failed to identify most positive cases, resulting in low F1 and MCC scores. While the *few-shot with guardrails* scenario still produced the highest scores, the introduction of guardrails in the prompt proved to be the most effective intervention, e.g., in the case of *Llama-3-8B*'s MCC score, for guiding the model toward a faithful and grounded answer.

In the optimal *few-shot with guardrails* scenario, *Llama-3-8B* and *Qwen2.5-7B* emerged as the best performing models. Instead, the *Ministral-8B* model performed better than others in both the *zero-shot* and *few-shot* settings, while *Mistral-Nemo* achieved the highest performance in the *zero-shot* prompt *with guardrails*. Smaller models like *Llama-3.2-1B* struggled significantly, highlighting their limitations in handling aggregated data about the log.

The evaluation results for the *events* dataset, shown in Table 4, report the framework's performance on *event-related* queries, where many LLMs achieved high performance, particularly in identifying all positive instances.

Indeed, a notable pattern in the *zero-shot* scenario is the LLMs' inclination towards high recall. Several models, namely *Mistral-7B-v0.3*, *Ministral-8B*, and *gpt-4o-mini*, achieved perfect recall out of the box. It is worth noting the impact of guardrails on the prompt for this dataset, with LLMs that initially struggled getting a significant performance boost. However, for some already high-performing models like Ministral-8B, the guardrails proved to be overly restrictive: when the model found retrieved context that was not an exact textual match for an element in the question, the strict guardrail led it to default to an 'I don't know' answer rather than inferring the correct one. This resulted in significant drops in performance because the models preferred to avoid potential guesses. Thus, while guardrails are generally beneficial, they can sometimes hinder the abilities of a model that has already learned to answer properly. The *few-shot with guardrails* prompt proved again optimal, as the few-shot examples helped maintain the high recall while the guardrails ensured high precision, leading to the better overall F1 and MCC scores.

In the latter prompting setting, *Ministral-8B* and *gpt-4o-mini* emerged as the best models: both achieved a high accuracy and MCC scores, balancing precision and recall and demonstrating a strong understanding of event-specific details within the retrieved context. Also *Llama-3-8B*, *Llama-3.1-8B*, and *Qwen2.5-7B* delivered reliable performance. As with other datasets, the smaller models struggled the most.

From the object perspective, Table 5 demonstrates that queries about specific *objects* and their attributes are the most straightforward for the framework to handle, as it is

**Table 3** Framework evaluation with various LLMs on the *global statistics* dataset across different prompt engineering scenarios. For each metric within each scenario, the best results are highlighted in bold, while the runner-ups are underlined

| LLM | Zero-Shot | | | | | | Zero-Shot with Guardrails | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** |
| *Llama-3-8B* | 0.65 | **1.00** | 0.31 | 0.47 | 0.43 | 0.65 | 0.90 | 0.93 | **0.87** | **0.90** | 0.80 | 0.90 |
| *Llama-3.1-8B* | 0.83 | 0.95 | 0.68 | 0.80 | 0.68 | 0.83 | 0.89 | 0.95 | 0.82 | 0.88 | 0.78 | 0.89 |
| *Llama-3.2-1B* | 0.53 | 0.60 | 0.15 | 0.24 | 0.08 | 0.52 | 0.53 | 0.53 | 0.57 | 0.55 | 0.07 | 0.53 |
| *Llama-3.2-3B* | 0.64 | 0.89 | 0.33 | 0.48 | 0.37 | 0.64 | 0.83 | 0.92 | 0.72 | 0.81 | 0.68 | 0.83 |
| *Mistral-7B-v0.2* | 0.52 | **1.00** | 0.04 | 0.08 | 0.15 | 0.52 | 0.80 | 0.97 | 0.61 | 0.75 | 0.64 | 0.80 |
| *Mistral-7B-v0.3* | 0.88 | 0.90 | 0.85 | 0.88 | 0.76 | 0.88 | 0.88 | 0.90 | 0.85 | 0.88 | 0.76 | 0.88 |
| *Mistral-Nemo* | 0.84 | 0.97 | 0.70 | 0.81 | 0.70 | 0.84 | **0.91** | 0.97 | 0.84 | **0.90** | **0.82** | **0.91** |
| *Ministral-8B* | **0.93** | 0.91 | **0.95** | **0.93** | **0.86** | **0.93** | 0.83 | 0.98 | 0.68 | 0.80 | 0.70 | 0.83 |
| *Qwen2.5-7B* | 0.79 | **1.00** | 0.58 | 0.74 | 0.64 | 0.79 | 0.88 | 0.98 | 0.78 | 0.87 | 0.78 | 0.88 |
| *gemma-2-9b* | 0.73 | 0.98 | 0.48 | 0.64 | 0.55 | 0.73 | 0.59 | **1.00** | 0.17 | 0.30 | 0.31 | 0.59 |
| *gpt-4o-mini* | 0.81 | 0.99 | 0.63 | 0.77 | 0.67 | 0.81 | 0.83 | 0.99 | 0.67 | 0.80 | 0.70 | 0.83 |
| | **Few-Shot** | | | | | | **Few-Shot with Guardrails** | | | | | |
| **LLM** | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* |
| *Llama-3-8B* | 0.74 | **1.00** | 0.48 | 0.65 | 0.56 | 0.74 | **0.90** | 0.89 | **0.92** | **0.90** | **0.80** | **0.90** |
| *Llama-3.1-8B* | 0.68 | 0.92 | 0.39 | 0.55 | 0.44 | 0.68 | 0.82 | 0.90 | 0.71 | 0.79 | 0.65 | 0.82 |
| *Llama-3.2-1B* | 0.48 | 0.44 | 0.15 | 0.22 | −0.06 | 0.48 | 0.49 | 0.49 | 0.54 | 0.52 | −0.02 | 0.49 |
| *Llama-3.2-3B* | 0.67 | 0.81 | 0.45 | 0.58 | 0.38 | 0.67 | 0.74 | 0.79 | 0.65 | 0.71 | 0.48 | 0.74 |
| *Mistral-7B-v0.2* | 0.53 | 0.79 | 0.09 | 0.16 | 0.14 | 0.53 | 0.60 | 0.87 | 0.23 | 0.36 | 0.29 | 0.60 |
| *Mistral-7B-v0.3* | 0.85 | 0.91 | 0.78 | 0.84 | 0.71 | 0.85 | 0.82 | 0.85 | 0.78 | 0.81 | 0.64 | 0.82 |
| *Mistral-Nemo* | 0.84 | **1.00** | 0.68 | 0.81 | 0.72 | 0.84 | 0.83 | **0.99** | 0.67 | 0.80 | 0.70 | 0.83 |
| *Ministral-8B* | **0.89** | 0.90 | 0.88 | **0.89** | **0.79** | **0.90** | 0.88 | 0.95 | 0.79 | 0.86 | 0.76 | 0.88 |
| *Qwen2.5-7B* | 0.88 | 0.97 | 0.78 | 0.87 | 0.77 | 0.87 | **0.90** | 0.98 | 0.81 | 0.89 | **0.80** | **0.90** |
| *gemma-2-9b* | 0.55 | 0.53 | **0.99** | 0.69 | 0.22 | 0.55 | 0.72 | 0.94 | 0.47 | 0.63 | 0.52 | 0.72 |
| *gpt-4o-mini* | 0.78 | 0.99 | 0.57 | 0.72 | 0.62 | 0.78 | 0.78 | **0.99** | 0.57 | 0.73 | 0.62 | 0.78 |

**Table 4** Framework evaluation with various LLMs on the *events* dataset across four different prompt engineering scenarios. For each metric within each scenario, the best results are highlighted in bold, while the runner-ups are underlined

| **LLM** | **Zero-Shot** | | | | | | **Zero-Shot with Guardrails** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** |
| *Llama-3-8B* | 0.83 | 0.84 | 0.81 | 0.82 | 0.66 | 0.83 | 0.89 | 0.86 | 0.94 | 0.90 | 0.79 | 0.89 |
| *Llama-3.1-8B* | 0.58 | 0.72 | 0.26 | 0.39 | 0.21 | 0.58 | 0.92 | 0.88 | 0.98 | 0.93 | 0.85 | 0.92 |
| *Llama-3.2-1B* | 0.49 | 0.33 | 0.01 | 0.01 | −0.03 | 0.50 | 0.50 | 0.00 | 0.00 | 0.00 | 0.00 | 0.50 |
| *Llama-3.2-3B* | 0.88 | 0.83 | 0.97 | 0.89 | 0.77 | 0.88 | 0.84 | 0.76 | **1.00** | 0.86 | 0.72 | 0.84 |
| *Mistral-7B-v0.2* | 0.50 | 0.00 | 0.00 | 0.00 | 0.00 | 0.50 | 0.92 | 0.88 | 0.97 | 0.92 | 0.84 | 0.92 |
| *Mistral-7B-v0.3* | 0.93 | 0.87 | **1.00** | 0.93 | 0.86 | 0.93 | 0.93 | 0.88 | **1.00** | **0.94** | 0.87 | 0.93 |
| *Mistral-Nemo* | 0.68 | 0.87 | 0.43 | 0.58 | 0.42 | 0.68 | 0.68 | 0.80 | 0.49 | 0.61 | 0.40 | 0.68 |
| *Ministral-8B* | 0.92 | 0.86 | **1.00** | 0.92 | 0.85 | 0.92 | 0.71 | 0.82 | 0.53 | 0.65 | 0.45 | 0.71 |
| *Qwen2.5-7B* | **0.95** | **0.96** | 0.95 | **0.95** | **0.91** | **0.95** | 0.93 | 0.88 | **1.00** | **0.94** | 0.87 | 0.93 |
| *gemma-2-9b* | 0.89 | 0.85 | 0.94 | 0.90 | 0.79 | 0.89 | 0.51 | 0.87 | 0.03 | 0.04 | 0.08 | 0.51 |
| *gpt-4o-mini* | 0.93 | 0.88 | **1.00** | 0.94 | 0.87 | 0.93 | **0.94** | **0.89** | **1.00** | **0.94** | **0.88** | **0.94** |
| | **Few-Shot** | | | | | | **Few-Shot with Guardrails** | | | | | |
| **LLM** | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* |
| *Llama-3-8B* | **0.94** | 0.89 | **1.00** | 0.94 | 0.88 | **0.94** | 0.94 | 0.89 | **1.00** | 0.94 | 0.88 | 0.94 |
| *Llama-3.1-8B* | 0.86 | 0.88 | 0.85 | 0.86 | 0.73 | 0.86 | 0.93 | 0.89 | 0.98 | 0.93 | 0.86 | 0.93 |
| *Llama-3.2-1B* | 0.53 | 0.66 | 0.14 | 0.23 | 0.11 | 0.53 | 0.78 | 0.75 | 0.82 | 0.79 | 0.56 | 0.78 |
| *Llama-3.2-3B* | 0.93 | 0.89 | 0.99 | 0.94 | 0.87 | 0.93 | 0.90 | 0.91 | 0.90 | 0.90 | 0.81 | 0.90 |
| *Mistral-7B-v0.2* | 0.58 | 0.77 | 0.24 | 0.36 | 0.23 | 0.58 | 0.60 | 0.70 | 0.36 | 0.48 | 0.24 | 0.60 |
| *Mistral-7B-v0.3* | 0.92 | 0.87 | **1.00** | 0.93 | 0.86 | 0.93 | 0.83 | 0.75 | **1.00** | 0.86 | 0.71 | 0.83 |
| *Mistral-Nemo* | 0.66 | **1.00** | 0.32 | 0.48 | 0.43 | 0.66 | 0.86 | **0.98** | 0.74 | 0.85 | 0.75 | 0.86 |
| *Ministral-8B* | 0.92 | 0.87 | 0.99 | 0.92 | 0.85 | 0.92 | **0.96** | 0.96 | 0.96 | **0.96** | **0.92** | **0.96** |
| *Qwen2.5-7B* | 0.72 | 0.81 | 0.56 | 0.66 | 0.45 | 0.72 | 0.92 | 0.88 | 0.98 | 0.93 | 0.85 | 0.92 |
| *gemma-2-9b* | 0.65 | 0.59 | 0.97 | 0.74 | 0.39 | 0.65 | 0.68 | 0.92 | 0.40 | 0.56 | 0.42 | 0.68 |
| *gpt-4o-mini* | **0.94** | 0.90 | **1.00** | **0.95** | **0.89** | **0.94** | **0.96** | 0.92 | **1.00** | **0.96** | 0.91 | **0.96** |

**Table 5** Framework evaluation with various LLMs on the *objects* dataset across four different prompt engineering scenarios. For each metric within each scenario, the best results are highlighted in bold, while the runner-ups are underlined

| **LLM** | **Zero-Shot** | | | | | | **Zero-Shot with Guardrails** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** |
| *Llama-3-8B* | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *Llama-3.1-8B* | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *Llama-3.2-1B* | 0.53 | 0.65 | 0.14 | 0.23 | 0.10 | 0.53 | 0.68 | 0.72 | 0.58 | 0.64 | 0.36 | 0.68 |
| *Llama-3.2-3B* | 0.85 | 0.86 | 0.85 | 0.85 | 0.71 | 0.85 | 0.95 | 0.98 | 0.92 | 0.95 | 0.90 | 0.95 |
| *Mistral-7B-v0.2* | 0.53 | **1.00** | 0.06 | 0.11 | 0.17 | 0.53 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *Mistral-7B-v0.3* | 0.99 | 0.99 | **1.00** | 0.99 | 0.99 | 0.99 | 0.99 | **1.00** | 0.98 | 0.99 | 0.97 | 0.99 |
| *Mistral-Nemo* | 0.99 | **1.00** | 0.99 | 0.99 | 0.99 | 0.99 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *Ministral-8B* | 0.98 | 0.96 | **1.00** | 0.98 | 0.96 | 0.98 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *Qwen2.5-7B* | 0.97 | 0.99 | 0.95 | 0.97 | 0.94 | 0.97 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *gemma-2-9b* | 0.59 | 0.99 | 0.18 | 0.31 | 0.32 | 0.59 | 0.85 | **1.00** | 0.69 | 0.82 | 0.73 | 0.85 |
| *gpt-4o-mini* | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| | **Few-Shot** | | | | | | **Few-Shot with Guardrails** | | | | | |
| **LLM** | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* |
| *Llama-3-8B* | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 0.99 | 0.99 | **1.00** | **1.00** | 0.99 | **1.00** |
| *Llama-3.1-8B* | 0.99 | **1.00** | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | **1.00** | 0.97 | 0.99 | 0.97 | 0.99 |
| *Llama-3.2-1B* | 0.50 | 0.49 | 0.08 | 0.14 | −0.01 | 0.50 | 0.72 | 0.83 | 0.55 | 0.66 | 0.46 | 0.72 |
| *Llama-3.2-3B* | 0.97 | 0.98 | 0.97 | 0.97 | 0.95 | 0.97 | 0.94 | **1.00** | 0.89 | 0.94 | 0.89 | 0.94 |
| *Mistral-7B-v0.2* | 0.68 | **1.00** | 0.35 | 0.52 | 0.46 | 0.68 | 0.92 | **1.00** | 0.84 | 0.91 | 0.85 | 0.92 |
| *Mistral-7B-v0.3* | 0.99 | 0.99 | **1.00** | **1.00** | 0.99 | **1.00** | **1.00** | 0.99 | **1.00** | **1.00** | **1.00** | **1.00** |
| *Mistral-Nemo* | 0.97 | **1.00** | 0.94 | 0.97 | 0.94 | 0.97 | 0.96 | **1.00** | 0.92 | 0.96 | 0.92 | 0.96 |
| *Ministral-8B* | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *Qwen2.5-7B* | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *gemma-2-9b* | 0.50 | 0.50 | **1.00** | 0.67 | 0.04 | 0.50 | 0.94 | **1.00** | 0.88 | 0.94 | 0.89 | 0.94 |
| *gpt-4o-mini* | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |

evidenced by the near-perfect scores achieved by a majority of the LLMs across almost all scenarios, even in the zero-shot baseline.

For this dataset, it becomes complex to meaningfully differentiate between best LLMs, i.e., *Ministral-8B*, *Qwen2.5-7B*, and *gpt-4o-mini*, as they all consistently deliver perfect results across the various prompting scenarios. However, the task remains a useful benchmark for identifying a baseline of capability, as again the smallest *Llama-3.2-1B* model struggled significantly.

While advanced prompting is not strictly necessary for the top-performing models on this dataset, the case of *gemma-2-9b* is interesting: from poor zero-shot performance, it reached perfect precision with guardrails and better overall performance in the few-shot prompt with guardrails, showing that even less-performing models can be guided to properly behave with the right prompting.

The evaluation on the *timestamps* dataset, detailed in Table 6, demonstrates the framework's ability in handling queries related to temporal information. Similar to the *objects* dataset, this task consists of factual data extraction from the context, where most capable LLMs perform with high accuracy.

In the optimal *few-shot with guardrails* configuration, *Ministral-8B* distinguishes itself by achieving perfect scores across all metrics, and other models such as *Llama-3-8B*, *Mistral-7B-v0.3*, and *Qwen2.5-7B* also delivered compelling results. Also in this case, the worst performance is registered by the smallest model, i.e., *Llama-3.2-1B*, which achieved a negative MCC score even with guardrails, and *gemma-2-9b*, showing inconsistent behavior without detailed task instructions and examples.

The prompting strategy proved to be the main factor in enhancing performance for this temporal dataset. In the baseline *zero-shot* scenario, many models demonstrated high precision, but sometimes at the cost of recall. The introduction of guardrails was highly effective at correcting this, e.g., for *Mistral-7B-v0.2*. Once again, the *few-shot with guardrails* strategy enabled the highest and most reliable performance with all the considered LLMs.

The outcomes of this multi-perspective evaluation suggest that, across all tested models and datasets, the *few-shot with guardrails* prompt engineering strategy regularly yielded the highest performance.

Although the quantitative results confirmed that the framework's effectiveness is not limited to a specific language model, a clear group of top performers emerged: *Llama-3-8B*, *Llama-3.1-8B*, *Ministral-8B*, *Qwen2.5-7B*, and *gpt-4o-mini* consistently ranked as the best across the various object-centric perspectives and evaluation metrics. To further examine the performance of these top five models and select a single best configuration, we generated a *box plot* visualizing their score distribution, as shown in Fig. 5. For this analysis, we focused on the MCC scores from the optimal *few-shot with guardrails* prompting scenario. We chose MCC as the representative metric because of its balance and robustness for classification tasks.

From this analysis, we can conclude that the *Ministral-8B* model performed the best, not only achieving the highest median MCC score, but also displaying a tight interquartile range. In light of this, such a LLM demonstrates a high degree of reliability, with its performance remaining near-perfect in the majority of test cases.

**Table 6** Framework evaluation with various LLMs on the *timestamps* dataset across four different prompt engineering scenarios. For each metric within each scenario, the best results are highlighted in bold, while the runner-ups are underlined

| **LLM** | **Zero-Shot** | | | | | | **Zero-Shot with Guardrails** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** | **Acc.** | **Prec.** | **Rec.** | **F1** | **MCC** | **AUC** |
| *Llama-3-8B* | 0.99 | 0.99 | 0.99 | 0.99 | 0.98 | 0.99 | **0.99** | 0.99 | **1.00** | **0.99** | 0.98 | **0.99** |
| *Llama-3.1-8B* | 0.99 | 0.99 | 0.99 | 0.99 | 0.98 | 0.99 | **0.99** | 0.99 | **1.00** | **0.99** | **0.99** | **0.99** |
| *Llama-3.2-1B* | 0.50 | 0.50 | 0.06 | 0.10 | 0.00 | 0.50 | 0.46 | 0.05 | 0.42 | 0.01 | −0.19 | 0.46 |
| *Llama-3.2-3B* | 0.91 | 0.93 | 0.89 | 0.91 | 0.83 | 0.91 | 0.95 | 0.94 | 0.95 | 0.95 | 0.90 | 0.95 |
| *Mistral-7B-v0.2* | 0.60 | **1.00** | 0.20 | 0.33 | 0.33 | 0.60 | **0.99** | 0.99 | **1.00** | **0.99** | **0.99** | **0.99** |
| *Mistral-7B-v0.3* | **1.00** | 0.99 | **1.00** | **1.00** | 0.99 | **1.00** | **0.99** | 0.99 | **1.00** | **0.99** | **0.99** | **0.99** |
| *Mistral-Nemo* | 0.96 | **1.00** | 0.92 | 0.96 | 0.92 | 0.96 | **0.99** | **1.00** | 0.98 | **0.99** | 0.98 | **0.99** |
| *Ministral-8B* | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **0.99** | **1.00** | 0.97 | **0.99** | 0.98 | **0.99** |
| *Qwen2.5-7B* | 0.97 | **1.00** | 0.95 | 0.97 | 0.95 | 0.97 | 0.98 | **1.00** | 0.96 | 0.98 | 0.96 | 0.98 |
| *gemma-2-9b* | 0.79 | **1.00** | 0.58 | 0.73 | 0.64 | 0.79 | 0.58 | **1.00** | 0.16 | 0.28 | 0.30 | 0.58 |
| *gpt-4o-mini* | 0.98 | **1.00** | 0.95 | 0.98 | 0.96 | 0.98 | **0.99** | **1.00** | 0.97 | **0.99** | 0.98 | **0.99** |
| | **Few-Shot** | | | | | | **Few-Shot with Guardrails** | | | | | |
| **LLM** | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* | *Acc.* | *Prec.* | *Rec.* | *F1* | *MCC* | *AUC* |
| *Llama-3-8B* | **1.00** | 0.99 | **1.00** | **1.00** | **0.99** | **1.00** | 0.99 | 0.98 | **1.00** | 0.99 | 0.98 | 0.99 |
| *Llama-3.1-8B* | 0.99 | 0.99 | **1.00** | 0.99 | **0.99** | 0.99 | 0.99 | 0.99 | **1.00** | 0.99 | **0.99** | 0.99 |
| *Llama-3.2-1B* | 0.47 | 0.43 | 0.17 | 0.24 | −0.07 | 0.47 | 0.72 | 0.89 | 0.51 | 0.65 | 0.50 | 0.72 |
| *Llama-3.2-3B* | 0.92 | 0.93 | 0.92 | 0.92 | 0.85 | 0.92 | 0.97 | 0.96 | 0.98 | 0.97 | 0.95 | 0.97 |
| *Mistral-7B-v0.2* | 0.73 | **1.00** | 0.45 | 0.62 | 0.54 | 0.73 | 0.84 | 0.98 | 0.68 | 0.81 | 0.70 | 0.84 |
| *Mistral-7B-v0.3* | 0.98 | 0.97 | **1.00** | 0.98 | 0.98 | 0.99 | 0.99 | 0.97 | **1.00** | 0.99 | 0.97 | 0.99 |
| *Mistral-Nemo* | 0.91 | **1.00** | 0.82 | 0.90 | 0.83 | 0.91 | 0.95 | **1.00** | 0.90 | 0.95 | 0.90 | 0.95 |
| *Ministral-8B* | 0.98 | 0.96 | **1.00** | 0.98 | 0.96 | 0.98 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| *Qwen2.5-7B* | 0.99 | **1.00** | 0.99 | 0.99 | **0.99** | 0.99 | 0.99 | **1.00** | 0.98 | 0.99 | 0.98 | 0.99 |
| *gemma-2-9b* | 0.50 | 0.50 | **1.00** | 0.67 | 0.00 | 0.50 | 0.87 | 0.99 | 0.75 | 0.85 | 0.77 | 0.87 |
| *gpt-4o-mini* | 0.96 | **1.00** | 0.92 | 0.96 | 0.92 | 0.96 | 0.97 | **1.00** | 0.95 | 0.97 | 0.95 | 0.97 |

Therefore, the framework's configuration employing *Ministral-8B* with the *few-shot with guardrails* prompt was selected as the optimal setup, and we employed it for the subsequent qualitative evaluation.

#### *Qualitative evaluation*

To address **RQ2**, we analyzed the questions submitted by participants to the framework about the P2P event log, as well as their responses to the administered survey. The complete log of the interactions between the users and the framework, reporting the question–answer pairs from the experiment, is available in the support material of the paper[11].

Regarding the questions posed, we found that they differed from those used in the quantitative evaluation dataset: participants' queries were generally more *discursive* and *higher-level*, as illustrated in Fig. 6. In particular, the participants valued the capacity demonstrated by the framework to integrate data on event–object relationships, connections among objects, their links to process activities, and descriptions of the overall execution into cohesive answers.

From the survey responses, participants reported that the framework's answers were usually well aligned with their questions. The cases in which the answer was judged as not aligned with the question are mainly related to the freedom we provided to the participants in experimenting with the framework. Indeed, in some cases, participants asked questions involving information not contained in the event log and marked the responses as not aligned, either because the framework correctly reported missing information or it answered only partially. All participants evaluated the responses positively in terms of clarity and considered the information sufficiently useful. They noted that the framework enhanced the accessibility and interpretability of object and event relationships and overall statistics, thereby improving the ability of non-technical users to understand object-centric logs.

The feedback also highlighted strengths, weaknesses, and potential areas for improvement. Strengths included ease of use, the ability to understand queries even when imperfectly formulated, clarity of responses, and responsiveness. Identified weaknesses and suggested improvements focused on extending current functionalities: adding guided interaction features to help refine questions with visual support from the event log, enabling more explicit responses when the framework cannot provide an answer, and supporting conversational memory across multiple questions to move beyond the current transactional interaction model.

## Threats to validity

The validity of this work is under a series of threats. A threat to internal validity is the *rapid evolution* of LLMs, which may render these results outdated over time. In response, we evaluated a range of both open-source and proprietary language models, demonstrating that the framework is not tied to any specific technology. Moreover, since the current results are already satisfactory, they are expected to further improve with the advent of more powerful and efficient models.

[11] The repository containing the code and datasets used in this paper is available at the following link: https://github.com/angelo-casciani/Conversational-OCEL2

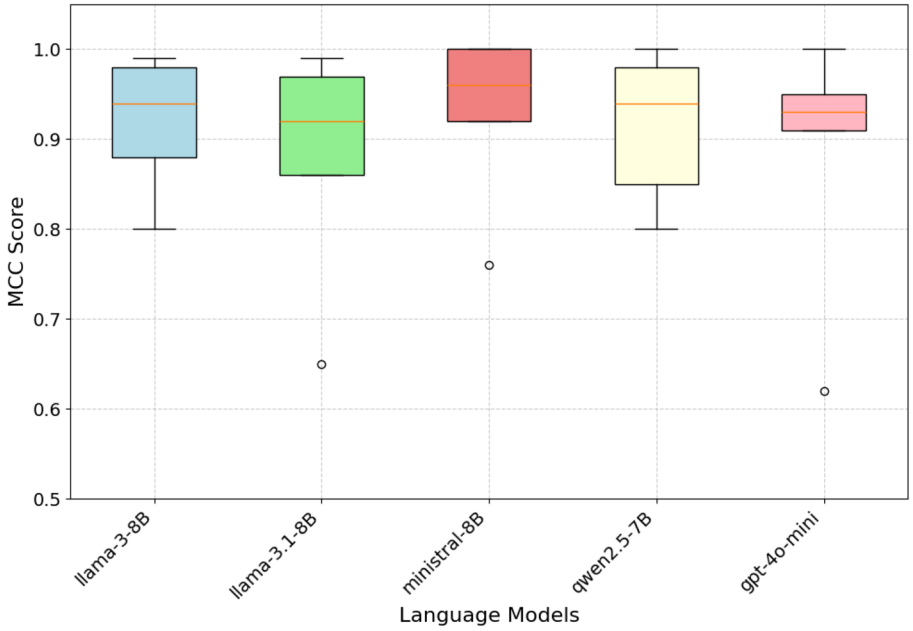


**Fig. 5** Box plots representing the framework's MCC performance for the *top five LLM* with the *few-shot prompt with guardrails* across all datasets

| | |
|---|---|
| *What is the maximum "Quantity" ordered for a "material" object and during which event did this order occur?* | *Can you describe the overall process underlying the event log?* |
| The maximum "Quantity" ordered for a "material" object is 100.0, and this occurred during event event:2727, where the object material is 246. | It appears that the event log is related to a business process involving procurement, production, and payment transactions within an organization.<br>The process involves multiple objects, such as materials, goods receipts, invoices, quotations, purchase orders, and payments.<br>Here's a high-level overview:<br>1. Material Procurement: Materials are procured through quotations and purchase orders, which involve interactions between procurement departments, procurement managers, and suppliers.<br>2. Goods Receipt: Goods are received from suppliers and recorded in the system. This step involves warehouse and finance/account departments.<br>3. Invoice Receipt: Invoices are received from suppliers and processed by finance/account departments.<br>4. Production: Materials are used to produce goods, which involves manufacturing departments.<br>5. Payment: Payments are made to suppliers for goods/services received.<br>This process likely involves various activities, such as ordering, receiving, invoicing, and paying. The event log provides insights into the timing and frequency of these events, allowing analysts to identify trends, patterns, and potential process issues. |
| *In which event was the "invoice_receipt" with maximum "Credit Amount (BSEG-WRBTR)" issued?* | |
| The maximum "Credit Amount (BSEG-WRBTR)" is 375050.0, and it corresponds to the event event:19684 and the object invoice receipt:1282. Therefore, the answer is: in event:19684, the "invoice receipt" with maximum "Credit Amount (BSEG-WRBTR)" was issued. | |
| *What are the object types that are manipulated by the activity "Perform Two-Way Match"?* | |
| It appears that the activity "Perform Two-Way Match" manipulates two object types: `invoice receipt` and `goods receipt`. These object types are mentioned in the `ocel:oid` fields throughout the different timestamps and activities. The "Two-Way Match" activity seems to be performing some kind of matching or reconciliation between these two object types. | |

**Fig. 6** Sample of the qualitative evaluation. Unlike the boolean questions in the dataset used for the quantitative evaluation, these queries posed by the study participants are more discursive and higher-level, requiring the language model to synthesize and reorganize information retrieved from the vector database

Instead, an external validity threat lies in the difficulty of reproducing the experiments due to the inherent *stochasticity* of LLMs when generating answers, even in response to identical prompts. To mitigate this issue, we provided the dataset used, detailed the prompt generation and composition, and formulated the evaluation as a binary classification task. In particular, we conducted an ablation study on the components of the prompt to assess the framework's sensitivity to minor variations. Additionally, we conducted *multiple runs* of the experiments to ensure the reliability of the reported outcomes.

Other limitations concern the qualitative evaluation. It was conducted with a relatively narrow participant group consisting of students and researchers from a single university.

While this choice facilitated controlled data collection and ensured participants had sufficient background knowledge to interact meaningfully with the framework, it does not reflect the primary target users of the system, i.e., process owners and managers in organizational settings. This limits the transferability and generalizability of the qualitative findings. Future work will therefore include structured user studies involving industry stakeholders to better capture their perspectives, needs, and challenges.

Moreover, while the overview on the possible questions presented to the user was needed to allow participants to ask meaningful queries even without prior knowledge of object-centric logs, it may have introduced a bias toward certain types of questions.

## Conclusion

In this paper, we introduced a *conversational framework* designed to facilitate *object-centric process analysis* employing *RAG-based LLMs*. Our framework enables users to query object-centric event logs in natural language, overcoming the limitations of traditional process mining techniques and query languages, which require programming expertise and often involve costly operations when handling multi-relational data. By preserving the interrelations between objects and events, our framework provides accurate and contextually relevant responses, making object-centric analysis more accessible, particularly for non-technical users.

We also introduced an evaluation dataset, derived from an OCEL 2.0 event log, intended as a benchmark for future assessments of conversational approaches applied to this log. The evaluation followed a twofold strategy. The quantitative assessment demonstrated that the framework can reliably generate grounded answers across multiple configurations of LLMs and prompting strategies under controlled experimental conditions. The subsequent qualitative study provided complementary insights into the perceived clarity, usefulness, and accessibility of the conversational interface. While these results are encouraging, the qualitative findings should be interpreted as indicative, given the limited and homogeneous participant sample and the preliminary guidance provided.

Future work will focus on strengthening the external validity of the framework through structured user studies involving industry practitioners to assess its impact in real-world decision-making contexts. Additionally, we plan to extend the framework by integrating business process model information with the execution aspects and render the preprocessing step independent of the event log format, enabling compatibility with XES event logs. Another promising direction for future research is the integration of the framework with Symbolic AI techniques to embed reasoning capabilities and enhance explainability, further improving the interpretability of object-centric insights.

**Acknowledgements**

We acknowledge financial support under the National Recovery and Resilience Plan (NRRP), M4C2I1.1, funded by the European Union - NextGenerationEU - aRtificial intElligence for Process Analytics (REPA), grant assignment decree No. 2022CJWPNA by the Italian Ministry of University and Research (MUR), the Sapienza project FOND-AIBPM, the PRIN 2022 project MOTOWN, and the NRRP MUR project PE0000013-FAIR. This work has been carried out while Angelo Casciani was enrolled in the National Doctorate on AI run by Sapienza University of Rome (CUP B53C23003500006, Mission 4 NRRP Generic Research Scholarship, Component 1).

**Author contributions**

Angelo Casciani carried out the experiments. All authors wrote and reviewed the manuscript.

**Funding**

Not applicable.

**Data availability**
The source code, event log, dataset, and instructions for replicating the experiments are available on GitHub at: https://github.com/angelo-casciani/Conversational-OCEL2. The dataset used for the quantitative evaluation is also available on Zenodo at: https://doi.org/10.5281/zenodo.18086659.

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

**Competing interests**
The authors declare no competing interests.

Received: 26 July 2025 / Accepted: 19 January 2026
Published online: 19 February 2026

**References**

Acampora G, Vitiello A, Stefano BND et al (2017) IEEE 1849: the XES standard: the second IEEE standard sponsored by IEEE computational intelligence society [society briefs]. IEEE Comput Intell Mag 12(2):4–8. https://doi.org/10.1109/MCI.2017.2670420

Agostinelli S, Del-Río-Ortega A, Goñi-Medina R et al (2025) Automating performance insights: suggesting and computing process performance indicators from event logs. In Advanced Information Systems Engineering - 37th International Conference, CAiSE 2025, Vienna, Austria, June 16-20, 2025, Proceedings, Part I, Lecture Notes in Computer Science, vol 15701, Springer, pp 221–237. https://doi.org/10.1007/978-3-031-94569-4_13

Barbieri L, Madeira ERM, Stroeh K et al (2023) A natural language querying interface for process mining. J Intell Inf Syst 61(1):113–142. https://doi.org/10.1007/S10844-022-00759-9

Barbieri L, Stroeh K, Madeira ERM et al (2024) An llm-based q&a natural language interface to process mining. In Process Mining Workshops - ICPM 2024 International Workshops, Lyngby, Denmark, October 14-18, 2024, Revised Selected Papers, Lecture Notes in Business Information Processing, vol 533, Springer, pp 5–17. https://doi.org/10.1007/978-3-031-82225-4_1

Bernardi ML, Casciani A, Cimitile M et al (2024) Conversing with business process-aware large language models: the BPLLM framework. J Intell Inf Syst 62(6):1607–1629. https://doi.org/10.1007/S10844-024-00898-1

Berti A, Jessen U, Park G et al (2025) Analyzing interconnected processes: using object-centric process mining to analyze procurement processes. Int J Data Sci Anal 20(2):475–497. https://doi.org/10.1007/S41060-023-00427-3

Berti A, Koren I, Adams JN et al (2024a) OCEL (object-centric event log) 2.0 specification. CoRR Abs/2403.01975. https://doi.org/10.48550/ARXIV.2403.01975

Berti A, Kourani H, van der Aalst WMP (2024b) Pm-llm-benchmark: evaluating large language models on process mining tasks. In Process Mining Workshops - ICPM 2024 International Workshops, Lyngby, Denmark, October 14-18, 2024, Revised Selected Papers, Lecture Notes in Business Information Processing, vol 533, Springer, pp 610–623. https://doi.org/10.1007/978-3-031-82225-4_45

Casciani A, Bernardi ML, Cimitile M et al (2024) Conversational systems for ai-augmented business process management. In Research Challenges in Information Science - 18th International Conference, RCIS 2024, Guimarães, Portugal, May 14-17, 2024, Proceedings, Part I, Lecture Notes in Business Information Processing, vol 513, Springer, pp 183–200. https://doi.org/10.1007/978-3-031-59465-6_12

Chirkova N, Troshin S (2021) Empirical study of transformers for source code. In: Spinellis D, Gousios G, Chechik M et al. (eds) ESEC/FSE 21: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, August 23-28, 2021, ACM, pp 703–715. https://doi.org/10.1145/3468264.3468611.

Dong Q, Li L, Dai D et al (2024) A survey on in-context learning. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024, Association for Computational Linguistics, pp 1107–1128. https://doi.org/10.18653/V1/2024.EMNLP-MAIN.64

Dubey A, Jauhri A, Pandey A et al (2024) The llama 3 herd of models. CoRR Abs/2407.21783. https://doi.org/10.48550/ARXIV.2407.21783

Esser S, Fahland D (2021) Multi-dimensional event data in graph databases. J Data Semant 10(1–2):109–141. https://doi.org/10.1007/S13740-021-00122-1

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 Information Systems Modeling - 25th International Conference, BPMDS 2024, and 29th International Conference, EMMSAD 2024, Limassol, Cyprus, June 3-4, 2024, Proceedings, Lecture Notes in Business Information Processing, vol 511, Springer, pp 22–31. https://doi.org/10.1007/978-3-031-61007-3_3

Fahland D (2022) Process mining over multiple behavioral dimensions with event knowledge graphs. In: Process mining handbook, lecture notes in business information processing, vol 448. Springer, pp 274–319. https://doi.org/10.1007/978-3-031-08848-3_9

Fahland D, Fournier F, Limonad L et al (2023) Why are my pizzas late? In Proceedings of the 2nd International Workshop on Process Management in the AI Era (PMAI 2023) co-located with 31st International Joint Conference on Artificial Intelligence (IJCAI 2023), Macao, S.A.R, August 19, 2023, CEUR Workshop Proceedings, vol 3569, CEUR-WS.org, pp 25–28. https://ceur-ws.org/Vol-3569/paper3.pdf

Gao Y, Xiong Y, Gao X et al (2023) Retrieval-augmented generation for large language models: a survey. CoRR Abs/2312.10997. https://doi.org/10.48550/ARXIV.2312.10997

Géron A (2022) Hands-on machine learning with Scikit-learn, Keras, and TensorFlow. O'Reilly Media, Inc
Ghahfarokhi AF, Park G, Berti A et al (2021) OCEL: a standard for object-centric event logs. In New Trends in Database and Information Systems - ADBIS 2021 Short Papers, Doctoral Consortium and Workshops: DOING, SIMPDA, MADEISD, MegaData, CAoNS, Tartu, Estonia, August 24-26, 2021, Proceedings, Communications in Computer and Information Science, vol 1450, Springer, pp 169–175. https://doi.org/10.1007/978-3-030-85082-1_16
Goossens A, Smedt JD, Vanthienen J (2024) Object-centric event logs: characteristics, comparative analysis and road map. In Business Process Management Forum - BPM 2024 Forum, Krakow, Poland, September 1-6, 2024, Proceedings, Lecture Notes in Business Information Processing, vol 526, Springer, pp 37–54. https://doi.org/10.1007/978-3-031-70418-5_3
Goossens A, Smedt JD, Vanthienen J et al (2022) Enhancing data-awareness of object-centric event logs. In Process Mining Workshops - ICPM 2022 International Workshops, Bozen-Bolzano, Italy, October 23-28, 2022, Revised Selected Papers, Lecture Notes in Business Information Processing, vol 468, Springer, pp 18–30. https://doi.org/10.1007/978-3-031-27815-0_2
Grohs M, Abb L, Elsayed N et al (2023) Large language models can accomplish business process management tasks. In: Business Process Management Workshops - BPM 2023 International Workshops, Utrecht, The Netherlands, September 11-15, 2023, Revised Selected Papers, Lecture Notes in Business Information Processing, vol 492, Springer, pp 453–465. https://doi.org/10.1007/978-3-031-50974-2_34
Günther M, Sturua S, Akram MK et al (2025) Jina-embeddings-v4: universal embeddings for multimodal multilingual retrieval. CoRR Abs/2506.18902. https://doi.org/10.48550/ARXIV.2506.18902
Han X, Hu L, Mei L et al (2020) A-BPS: automatic business process discovery service using ordered neurons LSTM. In 2020 IEEE International Conference on Web Services, ICWS 2020, Beijing, China, October 19-23, 2020, IEEE, pp 428–432. https://doi.org/10.1109/ICWS49710.2020.00063
Huang L, Yu W, Ma W et al (2025) A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ACM Trans Inf Syst 43(2):42:1–42:55. https://doi.org/10.1145/3703155
Jiang AQ, Sablayrolles A, Mensch A et al (2023) Mistral 7b. CoRR Abs/2310.06825. https://doi.org/10.48550/ARXIV.2310.06825
Kobeissi M, Assy N, Gaaloul W et al (2023) Natural language querying of process execution data. Inf Syst 116:102227. https://doi.org/10.1016/J.IS.2023.102227
Kojima T, Gu SS, Reid M et al (2022) Large language models are zero-shot reasoners. In Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022. http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html
Koroteev MV (2021) BERT: a review of applications in natural language processing and understanding. CoRR abs/2103.11943
Lewis P, Perez E, Piktus A et al (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Larochelle H, Ranzato M, Hadsell R et al. (eds) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html
Li G, de Murillas EGL, de Carvalho RM et al (2018) Extracting object-centric event logs to support process mining on databases. In Information Systems in the Big Data Era - CAiSE Forum 2018, Tallinn, Estonia, June 11-15, 2018, Proceedings, Lecture Notes in Business Information Processing, vol 317, Springer, pp 182–199. https://doi.org/10.1007/978-3-319-92901-9_16
Liss L, van der Aalst WMP (2025) Process area extraction by multilevel resource detection for object-centric process mining. In Business Process Management - 23rd International Conference, BPM 2025, Seville, Spain, August 31 - September 5, 2025, Proceedings, Lecture Notes in Computer Science, vol 16044, Springer, pp 197–214. https://doi.org/10.1007/978-3-032-02867-9_13
McKenzie IR, Lyzhov A, Pieler M et al (2023) Inverse scaling: when bigger isn't better. Trans Mach Learn Res. 2023.https://openreview.net/forum?id=DwgRm72GQF
Mikolov T, Chen K, Corrado G et al (2013) Efficient estimation of word representations in vector space. In 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings. http://arxiv.org/abs/1301.3781
Moctar-M'Baba L, Assy N, Sellami M et al (2023) Process mining for artifact-centric blockchain applications. Simul Model Pract Theory 127:102779. https://doi.org/10.1016/J.SIMPAT.2023.102779
Naveed H, Khan AU, Qiu S et al (2025) A comprehensive overview of large language models. ACM Trans Intell Syst Technol 16(5). https://doi.org/10.1145/3744746
OpenAI (2023) GPT-4 technical report. CoRR abs/2303.08774. https://doi.org/10.48550/ARXIV.2303.08774
Pasquadibisceglie V, Appice A, Malerba D (2024) LUPIN: a LLM approach for activity suffix prediction in business process event logs. In 6th International Conference on Process Mining, ICPM 2024, Kgs. Lyngby, Denmark, October 14-18, 2024, IEEE, pp 1–8. https://doi.org/10.1109/ICPM63005.2024.10680620
Rebmann A, Rehse J, van der Aa H (2022) Uncovering object-centric data in classical event logs for the automated transformation from XES to OCEL. In Business Process Management - 20th International Conference, BPM 2022, Münster, Germany, September 11-16, 2022, Proceedings, Lecture Notes in Computer Science, vol 13420, Springer, pp 379–396. https://doi.org/10.1007/978-3-031-16103-2_25
Rebmann A, Schmidt FD, Glavaš G et al (2025) On the potential of large language models to solve semantics-aware process mining tasks. Process Science 2(1):10. https://doi.org/10.1007/s44311-025-00019-3
Rebmann A, van der Aa H (2021) Extracting semantic process information from the natural language in event logs. In Advanced Information Systems Engineering - 33rd International Conference, CAiSE 2021, Melbourne, VIC, Australia, June 28 - July 2, 2021, Proceedings, Lecture Notes in Computer Science, vol 12751, Springer, pp 57–74. https://doi.org/10.1007/978-3-030-79382-1_4
Resinas M, Del-Río-Ortega A, van der Aa H (2023) From text to performance measurement: automatically computing process performance using textual descriptions and event logs. In Business Process Management - 21st International Conference, BPM 2023, Utrecht, The Netherlands, September 11-15, 2023, Proceedings, Lecture Notes in Computer Science, vol 14159, Springer, pp 266–283. https://doi.org/10.1007/978-3-031-41620-0_16
Riva F, Benvenuti D, Maggi FM et al (2023) An sql-based declarative process mining framework for analyzing process data stored in relational databases. In: Francescomarino CD, Burattin A, Janiesch C et al. (eds) Business Process Management Forum - BPM 2023 Forum, Utrecht, The Netherlands, September 11-15, 2023, Proceedings, Lecture Notes in Business Information Processing, vol 490, Springer, pp 214–231. https://doi.org/10.1007/978-3-031-41623-1_13.

Rivière M, Pathak S, Sessa PG et al (2024) Gemma 2: improving open language models at a practical size. CoRR Abs/2408.00118. https://doi.org/10.48550/ARXIV.2408.00118
Sahoo P, Singh AK, Saha S et al (2024) A systematic survey of prompt engineering in large language models: techniques and applications. CoRR Abs/2402.07927. https://doi.org/10.48550/ARXIV.2402.07927
Schönig S, Ciccio CD, Maggi FM et al (2016) Discovery of multi-perspective declarative process models. In Service-Oriented Computing - 14th International Conference, ICSOC 2016, Banff, AB, Canada, October 10-13, 2016, Proceedings, Lecture Notes in Computer Science, vol 9936, Springer, pp 87–103. https://doi.org/10.1007/978-3-319-46295-0_6
Schönig S, Rogge-Solti A, Cabanillas C et al (2016) Efficient and customisable declarative process mining with SQL. In Advanced Information Systems Engineering - 28th International Conference, CAiSE 2016, Ljubljana, Slovenia, June 13-17, 2016. Proceedings, Lecture Notes in Computer Science, vol 9694, Springer, pp 290–305. https://doi.org/10.1007/978-3-319-39696-5_18
van der Aalst WM (2023) Object-centric process mining: unraveling the fabric of real processes. Mathematics 11(12):2691
van der Aalst WMP (2022) Process mining: a 360 degree overview. In: Process mining handbook, lecture notes in business information processing, vol 448. Springer, pp 3–34. https://doi.org/10.1007/978-3-031-08848-3_1
van der Lee C, Gatt A, van Miltenburg E et al (2021) Human evaluation of automatically generated text: current trends and best practice guidelines. Comput Speech Lang 67:101151. https://doi.org/10.1016/J.CSL.2020.101151
Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. In: Guyon I, von Luxburg U, Bengio S et al. (eds) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
Vogelgesang T, Ambrosy J, Becher D et al (2022) Celonis PQL: a query language for process mining. In: Polyvyanyy A (ed) Process querying methods. Springer, pp 377–408. https://doi.org/10.1007/978-3-030-92875-9_13
Wang B, W, Chen F et al (2019) Evaluating word embedding models: methods and experimental results. CoRR abs/1901.09785. http://arxiv.org/abs/1901.09785
Yang A, Yang B, Zhang B et al (2024) Qwen2.5 technical report. CoRR Abs/2412.15115. https://doi.org/10.48550/ARXIV.2412.15115
Yeo H, Khorasani E, Sheinin V et al (2022) Natural language interface for process mining queries in healthcare. In IEEE International Conference on Big Data, Big Data 2022, Osaka, Japan, December 17-20, 2022, IEEE, pp 4443–4452. https://doi.org/10.1109/BIGDATA55660.2022.10020685

## Publisher's Note