

Original article

Orchestrating large language models to support medical process conformance checking

Giorgio Leonardi ^a, Stefania Montani ^{a,*}, Manuel Striani ^a, Alessandro Canessa ^b, Delfina Ferrandi ^b

^a Computer Science Institute, DiSIT, University of Piemonte Orientale, Viale Teresa Michel, 11, Alessandria, 15121, Italy

^b Integrated Laboratory of AI and Medical Informatics, DAIRI, SS. Antonio e Biagio e Cesare Arrigo Hospital, Alessandria, Italy

ARTICLE INFO

Keywords:

Medical processes
Large language models
Conformance checking
Stroke

ABSTRACT

Objective. Conformance checking in healthcare aims to verify whether patient care traces comply with clinical guidelines, but it typically requires formal, machine-interpretable guideline models, such as Computer-Interpretable Guidelines (CIGs), that are rarely available in practice.

Methods. This work presents a modular framework that leverages orchestrated Large Language Models (LLMs) to support medical conformance checking directly from unstructured data, without the need for CIGs. The proposed architecture combines multiple LLMs and auxiliary components to extract patient traces from clinical discharge letters, derive normative rules from textual clinical guidelines, formalize rules into executable scripts, and compute a Trace Conformance Indicator that quantifies conformance on the event log.

Results. The framework has been implemented and evaluated in the stroke care domain at the neurological ward of Alessandria hospital, where hundreds of patient traces were automatically extracted from hospital data. In particular, most of the available traces proved to be conformant to the 50 rules derived from the guideline.

Conclusion. Our work has demonstrated the feasibility of LLM orchestration in our domain, while, at the same time, verifying the good rule conformance checking results in Alessandria.

1. Introduction

Conformance checking is a fundamental activity in Process Mining (PM) [1] that assesses whether an organization's observed behavior complies with its prescribed processes. Observed behavior is typically captured in an event log, which consists of a collection of process traces [2]. Each trace records the sequence of activities carried out during routine operations, along with their timestamps. In the healthcare domain, a trace represents the ordered set of activities performed for a patient throughout a hospital stay. The behavior prescribed in this context is typically defined by a clinical guideline.

Conformance checking techniques rely on two main inputs: the event log and a formal, machine-readable specification of the normative process. This specification is commonly provided as a graphical control-flow model expressed in a dedicated modeling language, which can be converted into the well-known Petri Net formalism [3]. In healthcare settings, however, such formal representations are rarely available, as clinical guidelines are usually documented in extensive and complex

natural-language texts. Although methods for developing Computer-Interpretable Guidelines (CIGs) have been proposed [4], they require significant effort in terms of knowledge elicitation and modeling.

In this paper, we introduce a framework that orchestrates different Large Language Models (LLMs) to support medical conformance checking by validating patient traces directly on a textual guideline, without the need for an explicit CIG.

Our approach complies with a recent trend in Artificial Intelligence (AI) applications, where we are assisting to a shift away from reliance on ever-larger standalone models toward compound systems, in which multiple components are deliberately orchestrated to solve complex tasks [5]. These systems can integrate LLMs with auxiliary modules such as retrieval mechanisms, symbolic solvers, execution engines, and verification steps, and/or chain different LLMs to complete the task. Empirical evidence from state-of-the-art systems demonstrates that such structured compositions can outperform monolithic models by enabling multi-step reasoning, solution filtering, and interaction with external knowledge sources. In practice, techniques such as chained reasoning pipelines have

* Corresponding author.

E-mail addresses: giorgio.leonardi@uniupo.it (G. Leonardi), stefania.montani@uniupo.it (S. Montani), manuel.striani@uniupo.it (M. Striani), alessandro.canessa@ospedale.al.it (A. Canessa), dferrandi@ospedale.al.it (D. Ferrandi).

<https://doi.org/10.1016/j.neuri.2026.100294>

Received 20 March 2026; Received in revised form 27 August 2026; Accepted 27 August 2026

Available online 29 August 2026

2772-5286/© 2026 The Author(s). Published by Elsevier Masson SAS. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

become common, underscoring that system-level orchestration design choices often yield larger performance gains than additional model scaling alone (see, e.g., [6,7]).

Within this line of investigation, our contribution consists in the design, implementation and testing of a modular architecture, where:

1. a first LLM extracts patient traces from unstructured clinical discharge letters, and exports them in XES format¹ for PM analyses;
2. a second LLM extracts normative behaviors from a textual guideline in the form of rules;
3. a third LLM filters out the un-useful rules, i.e., those that cannot be applied to the given event log (e.g., because they do not deal with the hospitalization phase, or due to data unavailability);
4. a fourth LLM formalizes the rules into executable Python scripts;
5. a fifth LLM refines the formalized rules by eliminating noise and artifacts, as well as by correcting bugs;
6. a Python module ultimately computes the Trace Conformance Indicator ($\mathcal{TCT}$), a metric that measures the proportion of event log traces satisfying each formalized rule. This enables clinicians and healthcare administrators to rapidly detect conformance violations or operational issues in real-world event logs.

The overall architecture has been extensively tested in the domain of stroke care, where we were able to extract 463 patient traces from real discharge letters collected at Alessandria Hospital, and to assess their conformance towards one of the most recently published Italian stroke clinical guidelines [8]. Details are provided in the following sections. In particular, Section 2 reviews related work; Section 3 technically describes the architecture; Section 4 presents experimental results, while Section 6 is devoted to conclusions.

2. Related work

In this section, we present related work, focusing first on classical conformance checking approaches, then on the use of LLMs in Process Mining, and finally on LLM orchestration.

2.1. Conformance checking

In the Process Mining literature (see, for example, [9]), conformance checking techniques are commonly classified into three broad categories: *log replay methods*, *trace alignment methods*, and *constraint-based approaches*.

Log replay methods simulate the execution of each trace from the event log on the prescribed process model. When the normative model is represented as a Petri Net, token-based replay techniques are typically employed [10], allowing the generation of additional tokens to overcome deadlocks and complete the execution; conformance is then assessed based on the number of missing or remaining tokens. In more general settings, where the model is not a Petri Net, conformance can be quantified using fitness metrics [11], defined as the proportion of traces that can be successfully replayed with respect to the total number of traces.

Trace alignment methods identify deviations at the level of individual activities by detecting asynchronous moves, that is, steps where the trace and the model cannot advance simultaneously. Each deviation is associated with a cost, and an optimal alignment minimizing the overall cost is computed [12]. While this approach yields more fine-grained and precise diagnostics than log replay, it is computationally more demanding, as the number of possible alignments increases rapidly with the size of the event log.

Constraint-based conformance checking approaches (e.g., [13]) represent process behavior in terms of declarative rules and assess whether traces satisfy these constraints. The method proposed in this work falls

within this category, with the key novelty that, to the best of our knowledge, it is the first to leverage an LLM to automatically extract such rules from a normative specification expressed in free natural language, within a larger orchestrated architecture for conformance checking in medicine.

2.2. LLMs in process mining

Large Language Models (LLMs), including BERT [14] and Google's Gemini [15], have significantly expanded the state-of-the-art in Natural Language Processing (NLP) [16], opening up new possibilities for applications in Process Mining (PM).

An overview of recent research at the intersection of LLMs and PM is provided in the survey by Berti et al. [17].

According to the survey, on the one hand, LLMs have been used to automatically produce natural language text and explanations from event logs and process models [18,19], as well as from visual artifacts such as dotted charts [20]. To improve process interpretability, researchers have adopted techniques including abstraction [18] and sophisticated prompt engineering strategies [19]. LLMs have also been leveraged to support natural language querying of process data, for example to identify performance bottlenecks or detect fairness-related issues [20].

On the other hand, LLMs have been used to extract process elements and/or process models from textual documents, by leveraging prompt engineering, error correction, and code generation mechanisms [21]. The work in [22] adopts a LLM-based approach to extract event logs from a human-annotated dataset of nurse notes in the MIMICS-III dataset. The work in [23] explores the use of LLMs to derive both imperative and declarative process models from text and to evaluate process suitability for robotic process automation, achieving improvements over earlier NLP-based techniques that were confined to narrower tasks. Interactive refinement of the generated models is also supported in [21]. Beyond model extraction, LLMs have been applied to generate SQL queries for filtering and analyzing event logs stored in relational databases [24].

In contrast to prior work on declarative PM (see, e.g., [23]), our objective is more focused: we do not seek to discover a complete declarative process model, and the required knowledge acquisition effort is correspondingly lower (while, at the same time, we also work on event log extraction from natural text, as in [22]). Furthermore, among the existing literature, only [18] explicitly addresses LLM-assisted conformance checking; however, its results on medical datasets are limited, and it does not propose a method for deriving rules directly from domain knowledge.

2.3. LLM orchestration

A growing number of AI architectures are being proposed in the literature, that orchestrate different LLMs, as well as other components, to complete a reasoning task, rather than developing excessively complex single models. Focusing on healthcare applications, LLM-Synergy [25], for instance, is an ensemble learning framework that integrates outputs from multiple LLMs to improve performance on medical question-answering datasets. Always in medical question-answering, HealthBranches [26] first extracts structured clinical knowledge from medical literature, and then exploits it to score different LLMs' answers through LLM-as-a-Judge evaluation.

Orchestration can also take place in an agentic fashion: agentic AI denotes autonomous AI systems capable of defining objectives, planning and reasoning about actions, and carrying out complex tasks with little human intervention. Unlike traditional command-response models, these systems operate as independent agents that adapt to changing environments and learn from experience to accomplish their goals. Agentic AI typically integrates LLMs with additional tools and algorithms to coordinate multi-step workflows, enabling proactive problem solving. In healthcare, Rodella et al. [27], for instance, describe an agentic

¹ <https://www.xes-standard.org/>

AI workflow that integrates multiple specialized LLMs into a coordinated pipeline to automate the design of simulation scenarios. The system decomposes complex tasks (such as clinical narrative generation, diagnostic data creation, and educational debriefing) into sub-tasks handled by different AI agents, and these are orchestrated using advanced techniques like prompt chaining, parallelization, retrieval augmentation, and iterative refinement — demonstrating how orchestrated AI workflows can significantly improve efficiency and usability in medical applications.

Our approach is not (yet) designed as agentic AI (this line of research will possibly be considered in the future). However, it still addresses the goal of reducing the systematic biases that may emerge when fully relying on a single model, enhancing robustness and generalizability, as well as transparency. To the best of our knowledge, it represents the first example of a LLM orchestration architecture able to support medical process conformance checking, drastically reducing the knowledge acquisition bottleneck from natural language, working at the same time both on the formalization of the event log and on the formalization of the normative rules.

3. LLM-assisted conformance checking

In our architecture, illustrated in Fig. 1, we orchestrate a set of LLMs, as well as a final Python module, to realize a pipeline, which accomplishes the following steps:

1. *Trace Extraction*: by means of a first LLM (currently, Gemini 2.5 Flash [15]), we process discharge letters, and convert them into an event log, formalized as a file of XES traces;
2. *Rule Extraction*: by means of a second LLM (currently, NotebookLM [28]), we extract a set of semi-structured rules (in the IF-THEN form) from a textual clinical guideline;
3. *Rule Filtering*: by means of a third LLM (currently, Gemini 2.5 Flash [15]), we filter out those rules that cannot be applied to the available event log. In fact, some rules refer to guideline recommendations that are not meant to be applied in the hospitalization phase (e.g., they refer to prevention), or simply require patient data that were not routinely collected in our dataset (and thus they cannot be checked). Separating the rule filtering phase (which requires the provision of the event log to inform the LLM on what data are available) from the previous rule extraction phase allows us to obtain an initial rule set which is not biased by the current event log content, but only depends on the guideline;
4. *Rule Coding*: by means of a fourth LLM (currently, still Gemini 2.5 Flash [15]), we convert the filtered rules into executable Python scripts; when applied to a trace, the script will provide one of the following answers: *not applicable*, *conformant*, *not conformant*;
5. *Rule Refinement*: by means of a fifth LLM (currently, Gemini 3 Pro-Preview [29]) we improve the Python rules, by fixing possible coding bugs, and eliminating redundancy, noise and artifacts;
6. *Conformance Checking*: by means of a last Python module, we apply the refined rules to the event log, and compute a metric that we termed Trace Conformance Indicator ($\mathcal{TCI}$), which provides the percentage of conformant traces for each applicable rule.

Given a rule, $\mathcal{TCI}$ is defined as follows:

$$\mathcal{TCI} = \left(\frac{\text{conformant_traces}}{\text{total_applicable_traces}} \right) \times 100$$

where

- *total_applicable_traces* is the number of traces the rule at hand can be applied to (e.g., if the rule states the timing for thrombolysis, *total_applicable_traces* counts all and only the traces where thrombolysis was administered);

- *conformant_traces* is the number of traces where the rule at hand was applicable AND was correctly applied (e.g., if the rule states the timing for thrombolysis, *conformant_traces* counts all and only the traces where thrombolysis was administered at the correct time); $\text{conformant_traces} \leq \text{total_applicable_traces}$.

In order to extract traces from the discharge letters, we adopt a *few-shot prompting* technique [30,31], where, along with the task description, we provide examples of trace format and content (specifically XES tags, activity reference lists, activity names).

Fig. 2 illustrates the adopted prompt, highlighting the different strategies [30,31] we use. Specifically, we resort to:

- *role prompting*: we assert that the LLM is a medical PM expert, to force the model (which is a frozen one) to use already available domain knowledge, and an approach more focused to the task;
- *contextual prompting*: to build context, we provide the discharge letter, and an activity synonym reference list (i.e., a vocabulary). In particular, such a reference list had been learned by the LLM itself in a previous step of the work, where all the discharge letters were provided to the model; the list was then validated by medical experts. In the future we will consider to integrate a standard vocabulary, such as SNOMED-CT², along the lines described in [32], in a Retrieval Augmented Generation (RAG) [31,33] fashion;
- *system prompting*: we establish some general rules that the model has to follow when completing the task, such as respecting the XES syntax, use reference list terms in the traces, use Italian.

A similar strategy is used for prompting in the other phases of the work (i.e., rule extraction, rule filtering, rule coding and refinement; an additional example is provided in the Appendix (see Fig. A.5).

As regards the choice of the LLMs, it was basically dictated by the availability of Google tools within an educational agreement (Google Workspace for Education) active at our university. LLM settings (e.g., temperature) are reported in Table A.5 in the Appendix.

It is worth noting that we also tested Gemini for rule extraction, but it only provided a small rule set (16 rules), leading us to prefer NotebookLM for this task. NotebookLM, which extracted 161 rules (see Section 4), also proved more accurate with respect to GPT-5 *Thinking*, which we tested in our past work [34] (where we adopted a different event log), that only provided 28 rules. Possible further tests will be considered in the future.

4. Experiments

This section presents our experiments in the stroke domain. Stroke is a serious medical emergency that occurs when blood supply to the brain is disrupted, leading to the death of brain cells. This disruption can result from ischemia—caused by a blood clot or embolus that deprives brain tissue of oxygen and glucose—or from hemorrhage. During the acute stage, stroke poses a significant threat to life, and survivors frequently experience severe complications that may result in long-term or permanent disability. Stroke is the primary cause of disability among adults in the United States and Europe and is the second leading cause of death worldwide. In this work, we focus on the emergency stroke unit admission/hospitalization stage of stroke care.

Below, we first detail how the different steps of the orchestrated LLM pipeline were implemented in this field; then, we provide details of the validation; finally, we showcase our results. Note that every step was validated and its output was corrected (in case of need), before moving to the next step.

² <https://www.snomed.org/>

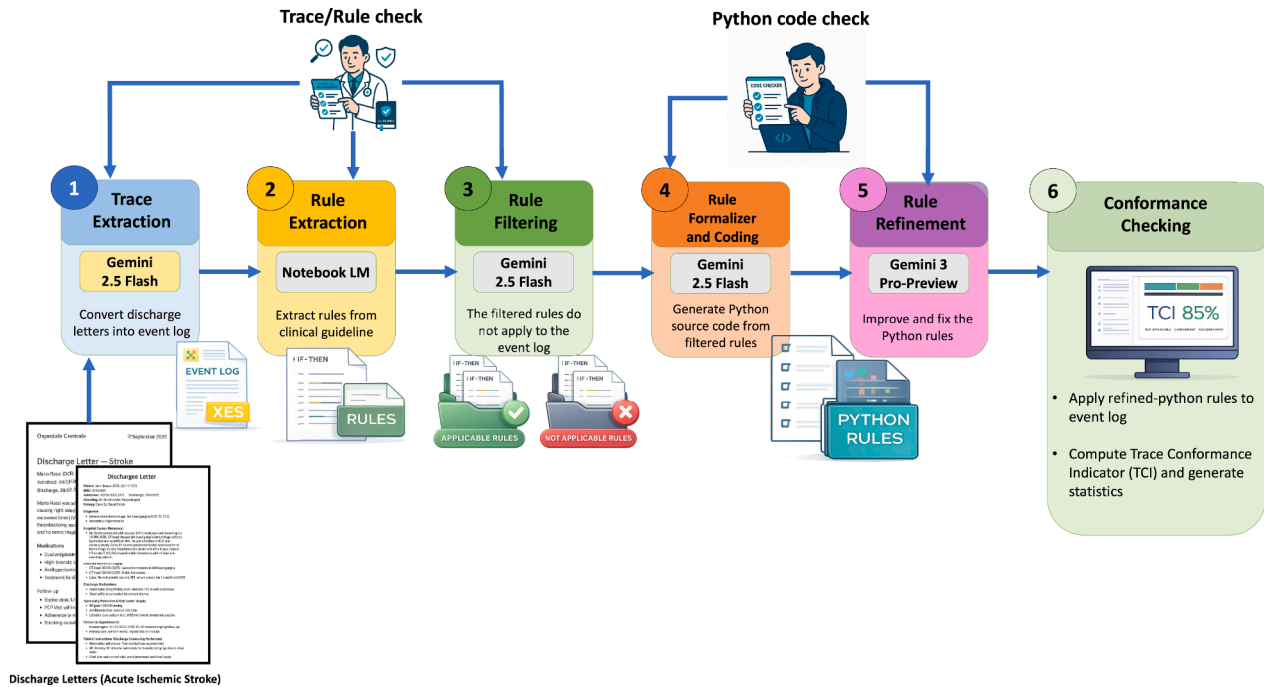


Fig. 1. Orchestrated LLM pipeline for guideline-based conformance checking.

4.1. Application of the pipeline steps to the stroke domain

4.1.1. Trace extraction

In our experiments, we considered 463 discharge letters of real stroke patients, cared at the neurology ward of Alessandria Hospital (Italy), over the years 2022–2024³. All the available letters were included, without making any selection on the basis of the stroke type or any other feature. 391 patients had ischemic stroke, 66 had haemorrhagic stroke, and 6 experienced both. Additional details about patients' characteristics can be found in Tables A.1 and A.2 in the Appendix.

The aim of the study was that of translating the letters into process traces, in order to automatically check the conformance of the implemented activities with respect to the Italian reference guideline, in a quality assessment perspective.

Discharge letters were anonymised before any processing, so that no link between the patient identity and any of his/her data could be reconstructed, according to the directions of the ethics committee that approved the study. Moreover, the LLMs were accessed exclusively through the institutional agreement with Google which is active at our University (Google Workspace for Education), not through consumer accounts. The agreement provides the guarantee that the data are kept locally, and are not used for training the models. The uploaded and generated contents are property of the institution and are subject to strict privacy protection standards, according to the EU GDPR.

Gemini 2.5 Flash extracted an event log of 463 items, composed of 47 activities on average, corresponding to a mean hospitalization duration of 10 days.

4.1.2. Rule extraction

NotebookLM extracted 161 rules from the Italian guideline [8], organized into 9 categories: (1) diagnosis and initial imaging (20 rules); (2)

acute phase management of ischemic stroke (19 rules); (3) acute phase management of hemorrhagic stroke (16 rules); (4) monitoring and complications (20 rules); (5) primary prevention (24 rules); (6) secondary prevention with antithrombotic therapy (15 rules); (7) secondary prevention with surgery (12 rules); (8) rehabilitation (19 rules); (9) special populations (16 rules).

4.1.3. Rule filtering

Filtering kept only the rules belonging to the first 4 categories, which are connected with emergency stroke unit admission/hospitalization, which was the focus of our study. Some additional rules were then filtered out, because the data needed for the check were never collected on our patients, and/or were never reported in the discharge letter. Overall, only 50 rules survived.

Table A.3 in the Appendix details how many rules were kept after the filtering step, category by category.

As an example, we show a rule which deals with the acute phase management of ischemic stroke: *IF a patient has an occlusion of the intracranial internal carotid artery, middle cerebral artery, M1-M2 segments, or anterior cerebral artery, A1 segment, AND is unresponsive or unable to undergo intravenous thrombolysis, THEN mechanical thrombectomy techniques are recommended within 6 hours of symptom onset.*

4.1.4. Rule coding and refinement

Fig. 3 reports part of the Python code corresponding to the rule mentioned above, as produced after the coding and refinement phases.

Note that refinement allows to automatically fix coding bugs, and eliminate redundancy, noise and artifacts. We made the choice to resort to Gemini 3 Pro-Preview to this end, as it exceeds the capacity of the other LLM models we used, and is thus suitable to identify intrinsically flawed items.

A rule may be *not applicable* to a given trace because the trace does not satisfy the rule premise, or because some key data are not recorded. In the example of Fig. 3, if the symptom onset timestamp is not recorded for a specific patient, the rule cannot be applied, because it is not possible to check whether thrombectomy took place within 6 h from the onset

³ Patient data cannot be made available, due to privacy reasons. Instead, our code is available at https://osf.io/qv36n/overview?view_only=7a5f968ecbf143b09a407e9205546b9e

You are an expert in clinical process mining.

Your task is to analyze a hospital discharge letter and generate a process trace in XES format, following the official standard as rigorously as possible.

Follow these rules strictly and without exceptions:**1. XES STRUCTURE RULES*:**

- The output must be a complete and valid XML file that starts with `<?xml version='1.0' encoding='UTF-8'?>`.
- The root tag must include the necessary extensions:
 - `<log xes:version='1.0' xmlns='http://www.xes-standard.org/'>`
 - `<extension name='Concept' prefix='concept' uri='http://www.xes-standard.org/concept.xesext'/>`
 - `<extension name='Time' prefix='time' uri='http://www.xes-standard.org/time.xesext'/>`
- All events must be in **strict chronological order** and contained inside a `<trace>` tag.
- Each event (`<event>`) must contain EXACTLY the following tags and key attributes:

- For the timestamp: `<date key='time:timestamp' value='...'/>`
- For the activity name: `<string key='concept:name' value='...'/>`
- For details: `<string key='note' value='...'/>`

2. RULES FOR ACTIVITY NAMES (concept:name)*:

- **LANGUAGE:** The value attribute must ALWAYS be in **Italian**.*.
- **GOLDEN RULE OF STANDARDIZATION (MANDATORY)*:**
For the value of `<string key='concept:name' value='...'/>`, you MUST use ONLY the STANDARD NAMES present in the reference list below. Analyze the letter text, identify an event, find its variant in the reference list, and use the corresponding STANDARD name. You are not allowed to invent new standard names if a mapping exists. The reference list has absolute priority.
- **RULE FOR ACTIVITIES NOT PRESENT IN THE LIST*:**
If an activity in the text does not match any variant in the reference list, create a new standard activity name. Use the structure of STANDARD names in the list (e.g., "Therapy Start...", "Neurological Assessment...", "CT Scan Execution...") and create a new name that is:
 - **Concise and Clear***: 'Noun + Adjective' or 'Verb + Object' (e.g., "Cardiology Visit").
 - **Consistent***: Same terminology and style as the list.
 - **In Italian and with Capital Initials***.

```
--- START OF ACTIVITY REFERENCE LIST ---
{standardization_rules}
--- END OF ACTIVITY REFERENCE LIST ---
```

FINAL REVIEW*: Before generating the output, reread every tag `<string key='concept:name' value='...'/>` and ensure its value complies with the rules.

Generate the XES trace for the following text:

```
--- START OF FILE ---
{letter_content}
--- END OF FILE ---
```

Legend

- Role prompting
- System prompting
- Contextual prompting
- Few-shot prompting

* Implicit zero-shot prompting

Fig. 2. Prompt template with color-coded prompting strategies. The original prompt (in Italian) has been translated into English for the sake of paper clarity.

itself. *Not conformant*, instead, means that the procedure was applied, but in a non-correct way with respect to the guideline directives (in the example, thrombectomy took place more than 6 h after the onset).

As it can be observed, the script also takes advantage of the activity synonym reference list to facilitate conformance checking of patient

traces (where synonyms of the medical terms might have been used - see, e.g., the *thrombectomy_terms* list in Fig. 3). Moreover, temporal constraints are properly calculated.

```

1 def rule_thrombectomy(trace: list[dict]) -> str:
2     def _search_term(event: dict, terms: list) -> bool:
3         for k, v in event.items():
4             val = str(v).lower()
5             for t in terms:
6                 if t.lower() in val:
7                     return True
8         return False
9
10    occlusion_ica_terms = ['Occlusion of the intracranial internal carotid artery',
11                          'Intracranial ICA near-occlusion', ...]
12    thrombectomy_terms = ['Mechanical thrombectomy', 'Endovascular thrombectomy', ...]
13    onset_terms = ['stroke onset', 'onset', 'stroke detection', ...]
14    ...
15
16    for event in trace:
17        if not has_occlusion_ica and _search_term(event, occlusion_ica_terms):
18            has_occlusion_ica = True
19            ...
20        if onset_ts is None and _search_term(event, onset_terms):
21            current_ts = parse_timestamp(event.get('time:timestamp'))
22            if current_ts:
23                onset_ts = current_ts
24        if _search_term(event, trombolysis_iv_terms):
25            iv_trombolysis_administered = True
26            ...
27    if onset_ts is None and len(trace) > 0:
28        onset_ts = parse_timestamp(trace[0].get('time:timestamp'))
29    ...
30
31    if onset_ts is None:
32        return 'NOT APPLICABLE'
33
34    if mechanical_thrombectomy_performed:
35        time_delta_minutes = get_time_delta_minutes(onset_ts, mechanical_thrombectomy_ts)
36
37        if time_delta_minutes is not None and time_delta_minutes <= 360:
38            return 'CONFORMANT'
39        else:
40            return 'NOT CONFORMANT'
41    else:
42        return 'NOT CONFORMANT'

```

Fig. 3. Part of the Python code generated by Gemini for an example rule. In the figure, only some sections of the code are shown, for the sake of simplicity, and terms in the activity synonym reference list have been translated into English.

4.1.5. Conformance checking

Conformance checking then took place, by calculating $\mathcal{TCI}$ on each of the 50 surviving rules. Results are provided in [Section 4.3](#).

4.2. Validation of pipeline steps

4.2.1. Trace extraction

In the current work, a randomly extracted subsample (20%) of the traces generated by Gemini 2.5 Flash was checked and validated by two medical experts, who independently analyzed the same traces.

The physicians verified the coherence between the discharge letter information and the trace, the correctness of activity names (taking into account the synonym dictionary), the correctness of the timestamps (when available) and/or of the temporal ordering of the activities.

All the checked traces were judged as correct and usable in their current form. Inter-rater agreement was 100%. Therefore, a second human-in-the-loop iteration, asking the LLM to refine the output, was not required. Such an iterative approach can however be implemented in case of need.

Limitations of the current validation on trace extraction are discussed in [Section 5](#).

4.2.2. Rule extraction

Our medical collaborators checked the rules extracted by the LLM, focusing especially on the first 4 categories, which are connected with the hospitalization phase.

Rules are expressed as simple IF-THEN statements, as shown in [Section 4.1.3](#), and were easy and fast to check by the human experts. The rules were deemed both semantically accurate — reflecting the medical knowledge in the guideline — and clinically relevant, as their application is necessary for delivering correct and timely patient care. Overall, a good synthesis capability (with respect to the original, quite verbose textual guideline) was demonstrated by the LLM, with the exception of the secondary prevention rules, which did not differ much from synthetic guideline versions that are usually available to physicians for quick reference. It is however worth noting that secondary prevention rules do not relate to acute phase hospitalization, and were thus not considered in the further steps of our experiments.

On a positive note, “less crucial” information (e.g., summaries of clinical trial outcomes, that the guideline reports to support recommendations) was properly left out, and the extracted rules were free from hallucinations.

On a negative one, however, 3 rules were considered incomplete with respect to the original guideline content. Moreover, while 4 rules were devoted to the diagnosis of subarachnoid hemorrhage, the diagnosis of cerebral hemorrhage was not treated in depth.

In order to solve these issues, we have implemented a human-in-the-loop strategy, where the physician identified the problems, and (with the help of a computer science expert) refined the prompt. In the case of cerebral hemorrhage rules, for instance, the prompt instructed the LLM to focus on such a diagnosis. In the next iteration, the LLM was thus asked to provide a more complete/correct rule set. In our current experiments, 2 overall iterations were enough to reach a satisfactory result.

4.2.3. Rule filtering

In order to verify that the automatic rule filtering step did not introduce any bias or error, the expert physicians reviewed the 50 rules, and approved the selection. No rule was manually re-inserted. See however Section 5 for a discussion on the limitations of our work as regards possible data incompleteness.

4.2.4. Rule coding and refinement

After the coding and refinement phase, validation of every Python rule was completed by a medical expert, side by side with a computer scientist: the computer scientist verified the technical and syntactic correctness of the code, and helped the physician identify the semantic content within the Python script, allowing him/her to make a final check on the adherence of the code to the rule content. No significant issues were identified at this time. A test on a sample of 50 artificially generated traces, with known (expert-generated) conformance labels, was also conducted in this phase: rules provided the correct label in 100% of the cases. We thus proceeded towards the application on real patients data.

4.3. Results

Fig. 4 shows a summary of our experimental results on conformance checking at the neurological ward of Alessandria Hospital. Table A.4 in the Appendix details the TCI value rule by rule. As it can be seen, most rules reached very high TCI values, testifying a robust performance for Alessandria, when focusing on emergency stroke unit admission/hospitalization, and referring to the patients admitted in the period 2022–2024.

Despite the overall good results, the (few) non conformance examples show that there is some space for further improvements. In the following, as an example, we report our investigation on two specific rules, one with a medium TCI value, and the other with a very low one.

The first rule is expressed as follows: “If a patient has cerebral hemorrhage, THEN it is suggested to start treating their high blood pressure and reversing any anticoagulant therapies as soon as they arrive at the emergency room”. Such a rule obtained a TCI value of 62%. In most cases, non conformance was due to a delay to therapy administration, which was managed only in the neurological ward, and not in the emergency room, testifying a suboptimal communication between the two wards. In a specific example, however, looking at the trace and at the corresponding discharge letter, we were able to identify a delayed diagnosis: the patient was initially treated as the victim of a home accident, that caused the haemorrhage, and only later it was understood that an ischemic stroke had determined the accident itself, depicting a situation which was more complex than a simple coordination bottleneck.

We also investigated a second rule, expressed as follows: “If a patient has ischemic stroke AND hyperthermia, THEN pharmacological correc-

tion of hyperthermia is indicated (preferably with paracetamol), maintaining the temperature below 37 °C”. This rule totalized a TCI value of only 6%. After analyzing the not conformant traces, we found that all the patients involved suffered from hyperthermia, but their treatment was different from paracetamol. Interestingly, however, most of these patients underwent a diagnostic path, which allowed physicians to identify the precise cause of hyperthermia, leading to a different, more proper therapeutic strategy. In particular, 69% of the traces are related to patients who had contracted bacterial infections, and were thus treated with antibiotics; 6% of the traces relate to Covid-19-infected patient, who were treated with antivirals and anti-inflammatories. While these examples technically represent non conformances, in fact they were correct adaptations of the therapeutic strategy to the specific situations, with the objective of tackling the cause of the problem, instead of just reducing the symptoms (as paracetamol would do). The few remaining cases refer to patients having non-problematic hyperthermia (brief episodes resolved spontaneously), or to patients developing hyperthermia just before discharge or transfer to other wards, probably treated at destination.

In conclusion, our tool enabled us to identify and examine the non conformance examples experienced in Alessandria, highlighting some existing issues (e.g., a reduced coordination between the emergency room and the neurological ward), but also shedding light on cases which are only apparently suboptimal.

Our architecture is non-monolithic, and, by design, it takes advantage of a set of sequential steps, which resort to models that are strong at specific tasks (e.g., NotebookLM is strong at extracting information from documents, Gemini is stronger at generating code). For the sake of completeness, however, we also ran an “ablation study”, where a single LLM was asked to complete all the pipeline, instead of resorting to different models for the different steps. In particular, we exploited NotebookLM end to end. The number of conformant rules dropped, but this was not the main issue, on our opinion. Rather, the weakest point was the impossibility of realizing a human-in-the-loop approach, and the total absence of control over the outputs of the single steps. For instance, NotebookLM converted the extracted natural language rules into a pseudo-code (partly Python, partly natural language), and no other model (nor human) was involved for correction and refinement. NotebookLM itself then built a series of functions to make the rules executable, and completed the pipeline. The overall performance result is weak and, being out of control, is extremely debatable, and completely lacks transparency. It is worth noting that a less monolithic ablation study, where the same LLM is adopted to cover all the pipeline steps, but interleaved with possible human interventions and corrections, would not add value to the work, in our opinion. Indeed, recent research [35] on LLMs shows that no single model consistently performs best across heterogeneous tasks, and that exploiting complementary strengths of different models typically outperforms relying on a single model for all tasks. Besides improving task-specific reliability, moreover, a modular design avoids making the overall system dependent on the weaknesses and failure modes of a single LLM and allows individual components to be independently evaluated, replaced, or upgraded as better models become available.

5. Limitations

We acknowledge that, in the current version, our work is affected by some limitations:

- first of all, our medical experts checked only 20% of the generated traces, randomly selected; in the future, it would be helpful to possibly involve more physicians, and complete the verification of all the available traces in a reasonable period of time;
- some rules were not applicable because the data needed for the check were not reported in the discharge letter. In the future, we may con-

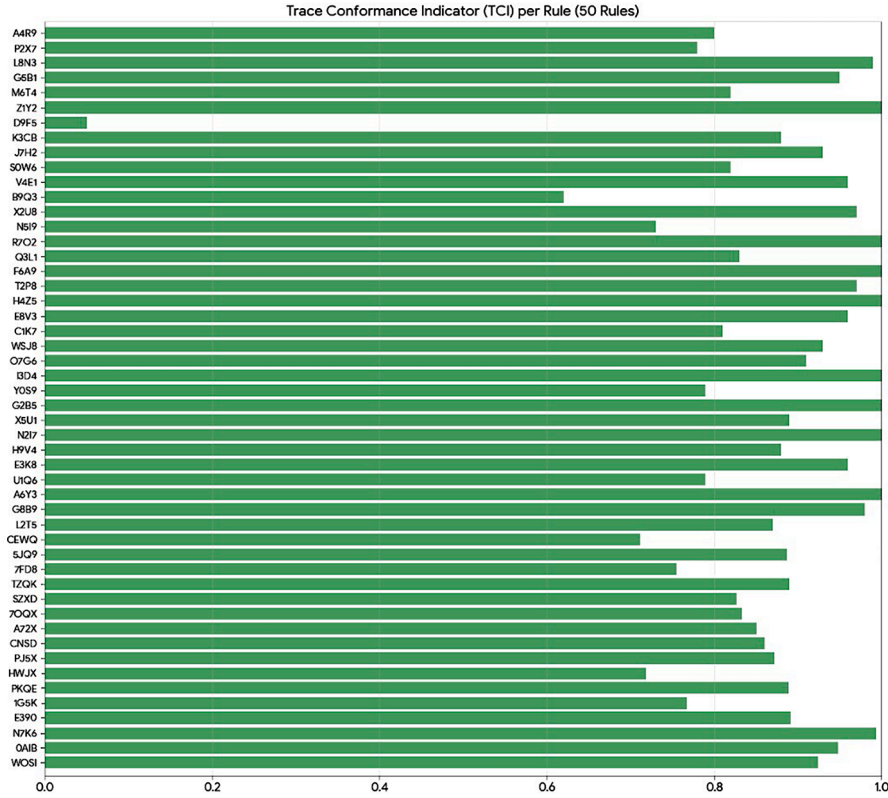


Fig. 4. Conformance analysis results.

sider the opportunity to integrate the discharge letter information with other hospital data sources, for the sake of completeness;

- on the same note, additional hospital data might provide more precise temporal information, often missing in the letters; by now, in many cases we had to simply consider the relative ordering of the actions, without having evidence of the precise timestamp;
- finally, in the current work we focused on the emergency stroke unit admission/hospitalization phase. While this was a deliberate choice, in the future it would be interesting to assess the conformance of Alessandria Hospital towards other sections of the guideline.

6. Conclusion

In this paper, we have proposed an architecture that integrates multiple LLMs and other supporting components to: (1) extract patient traces from clinical discharge letters, (2) infer normative rules from a textual clinical guideline, (3) formalize the relevant rules into executable representations, and (4) compute the TCI indicator, to measure trace conformance, supporting quality assessment without requiring a time-consuming formalization of the guideline itself.

Our contribution relies on the recent AI trend that moves towards system-level orchestration design, and, to the best of our knowledge, it is the first one able to cover the four tasks listed above, in an automatic and integrated way. The choice of relying on an orchestrated LLM pipeline instead of implementing a monolithic model, and the choice of validating the pipeline steps one by one, have allowed us to reduce bias and localize errors, and to provide full transparency about intermediate results. An end-to-end model would have offered no comparable intermediate control or explainability.

Our experiments within the stroke management domain have showcased the real-world usability of the approach, while at the same time verifying the good rule conformance checking results at the neurological ward of Alessandria Hospital (by now, limited to the guideline emergency stroke unit admission/hospitalization phase, and referring to the patients admitted in the period 2022–2024). Focusing on these experimental conditions, no major issues were identified, but a possible organizational bottleneck was found.

Collaboration with medical specialists, who verified the consistency of the extracted traces and rules with discharge letters and with established medical knowledge, has ensured the semantic validity of the current version of our approach.

Nevertheless, we consider it valuable to evaluate trace/rule extraction testing alternative LLMs, particularly models trained on healthcare-specific data, such as Google's Med-Gemini [36], which might derive even clearer and accurate information, potentially reducing the effort required during expert validation.

We also aim to advance toward the (partial) automation of the trace/rule verification process. To this end, we plan to explore the research paradigm known as LLM-as-a-Judge [37], in which LLMs are used to evaluate the outputs of other LLMs on complex tasks. Although human assessment remains the gold standard, it is often time-consuming and expensive; in contrast, LLM-based evaluation can provide faster feedback and more consistent criteria, assuming that a well-defined evaluation framework is in place. While the adoption of LLM-as-a-Judge methods must be approached with caution in medical contexts, such techniques could nonetheless accelerate validation by assigning an initial quality score, with selected cases subsequently reviewed by human experts when necessary.

We also plan to take care of the limitations listed in [Section 5](#).
Finally, in the future we plan to rely on the current framework to develop an education facility, testing the capability of non-expert physicians in learning the guideline and identifying non-conformant traces with respect to it, and supporting them towards the acquisition of a stronger competence in the field.

Ethics committee approval

The study was approved by Comitato Etico Territoriale Interaziendale AOU Maggiore della Carità di Novara (protocol code 216/CE and date of approval 19 March 2025).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Table A.1
Patients distribution by Length Of Stay (LOS) band.

LOS band	% pop.
1–3 days	20%
4–7 days	30%
8–14 days	35%
15–21 days	15%
>21 days	0.0%
Total	100%

Table A.2
Patients severity distribution as a percentage of cases with recorded *National Institutes of Health Stroke Scale* (NIHSS). Note that NIHSS is not available for all patients, or is available only at discharge.

Severity band	Adm. <i>n</i>	Adm. %	Disch. <i>n</i>	Disch. %
0–4, (minor)	94	28.4%	180	52.5%
5–15, (moderate)	139	42.0%	106	30.9%
16–20, (moderate-severe)	47	14.2%	38	11.1%
21–42, (severe)	51	15.4%	19	5.5%
Total, recorded	331	100.0%	343	100.0%

Table A.3

Rule filtering by guideline section.

Guideline section	Total rules	Applicable rules	Rationale for the excluded rules
Diagnosis and Initial Imaging	20	13	Excluded rules rely on data not recorded in the discharge letters.
Acute Phase – Ischaemic Stroke	19	14	Excluded rules rely on data not recorded in the discharge letters.
Acute Phase – Haemorrhagic Stroke	16	6	Excluded rules rely on data not recorded in the discharge letters.
Monitoring and Complications	20	17	Excluded rules rely on data not recorded in the discharge letters.
Primary Prevention	24	0	All rules are excluded because they target primary prevention in the pre-stroke population, such as lifestyle, diet, and contraceptives; they do not refer to emergency stroke unit admission or hospitalization.
Secondary Prevention – Antithrombotic Therapy	15	0	All rules are excluded because they target secondary prevention; they do not refer to emergency stroke unit admission or hospitalization.
Secondary Prevention – Surgery	12	0	All rules are excluded because they target secondary prevention; they do not refer to emergency stroke unit admission or hospitalization.
Rehabilitation	19	0	All rules are excluded because they target rehabilitation; they do not refer to emergency stroke unit admission or hospitalization.
Special Populations and Rare Causes	16	0	All rules are excluded because they target populations which are absent in our dataset.
Total	161	50	–

Prompt for rule extraction from the stroke guideline.

****Role:**** Act as a medical informatics expert specialized in clinical documentation analysis and knowledge extraction.

****Task:**** Read and analyze the provided text, extracted from an official guideline for stroke management, and extract all normative rules in a logical and structured format.

****Output requirements:****

1. ****“IF... THEN...” semantics****: Each rule must be clearly expressed using the conditional logic “IF [condition/s]... THEN [action/recommendation]”.
2. ****Grouping by Topic****: Organize the extracted rules into clear and relevant categories (e.g., “Initial Diagnosis and Imaging”, “Intravenous Thrombolysis”, “Endovascular Treatment”, “Secondary Prevention”, etc.).
3. ****Atomicity****: Each rule must be atomic, encoding a single constraint or recommendation. If a sentence contains multiple rules, split them into separate rules.

****Desired Output Example:****

****Category: Intravenous Thrombolysis****

****Rule 1:**** IF a patient presents with an acute ischemic stroke AND treatment can be initiated within 4.5 hours of known symptom onset, THEN they should be considered for thrombolysis with alteplase.

****Rule 2:**** IF a patient has been treated with thrombolysis, THEN an anti-platelet agent should be started after 24 hours, unless contraindicated and after ruling out significant hemorrhages.

****Final Instructions:****

Analyze the following text and generate the complete list of rules by strictly following the instructions and format indicated above.

****Text to Analyze**** """ [INSERT TEXT HERE] """

Fig. A.5. Prompt used to elicit atomic IF-THEN rules from the guideline text. The original prompt (in Italian) has been translated into English for the sake of paper clarity.

Table A.4Rule Conformity for each rule and $\mathcal{TCI}$ Performance Metrics.

Rule Code	Conformant traces	Non-conformant traces	Non-applicable traces	Applicable traces	$\mathcal{TCI}$ (Conformant/Applicable)
A4R9	365	87	11	452	0,81
P2X7	221	64	178	285	0,78
L8N3	212	2	249	214	0,99
G6B1	300	17	146	317	0,95
M6T4	37	8	418	45	0,82
Z1Y2	452	0	11	452	1,00
D9F5	6	98	359	104	0,06
K3CB	360	50	53	410	0,88
J7H2	417	25	21	442	0,94
S0W6	372	82	9	454	0,82
V4E1	202	8	253	210	0,96
B9Q3	9	8	446	17	0,53
X2U8	28	1	434	29	0,97
N5I9	328	120	15	448	0,73
R7O2	88	0	375	88	1,00
Q3L1	5	1	457	6	0,83
F6A9	457	0	6	457	1,00
T2P8	326	9	128	335	0,97
H4Z5	370	0	93	370	1,00
E8V3	185	9	269	194	0,95
C1K7	364	88	1	452	0,81
WSJ8	83	5	375	88	0,94
O7G6	419	33	11	452	0,93
I3D4	2	0	461	2	1,00
Y0S9	83	22	358	105	0,79
G2B5	3	0	460	3	1,00
X5U1	275	28	160	303	0,91
N2I7	5	0	458	5	1,00
H9V4	7	1	455	8	0,88
E3K8	204	10	249	214	0,95
U1Q6	116	31	316	147	0,79
A6Y3	417	2	44	419	1,00
G8B9	104	2	357	106	0,98
L2T5	76	11	376	87	0,87
CEWQ	276	170	17	446	0,62
5JQ9	81	8	374	89	0,91
7FD8	315	100	48	415	0,76
TZQK	213	25	225	238	0,89
SZXD	379	76	8	455	0,83
7OQX	260	49	154	309	0,84
A7ZX	255	44	164	299	0,85
CNSD	168	28	267	196	0,86
PJ5X	74	11	378	85	0,87
HWJX	216	85	162	301	0,72
PKQE	182	23	258	205	0,89
1G5K	347	101	15	448	0,77
E390	168	21	274	189	0,89
N7K6	83	1	379	84	0,99
OAIB	120	7	336	127	0,94
WOSI	416	32	15	448	0,93

Table A.5

LLM models and main generation settings used in the pipeline.

LLM name	Main model settings
gemini-2.5-pro Python API	Default API settings were used: <code>temperature = 1.0</code> , <code>max_output_tokens = 65,536</code> , <code>top_p = 0.95</code> , and <code>top_k = 40</code> . The random seed was not set; therefore, the output is non-deterministic.
gemini-2.5-flash Python API	Default API settings were used: <code>temperature = 1.0</code> , <code>max_output_tokens = 65,536</code> , <code>top_p = 0.95</code> , <code>top_k = 40</code> , and <code>response_mime_type = text/plain</code> . The random seed was not set; therefore, the output is non-deterministic. Retry policy: <code>max_retries = 2</code> .
NotebookLM Python API	Generation parameters are not exposed programmatically: the NotebookLM API does not allow inspecting or setting temperature, token limit, <code>top-p</code> , <code>top-k</code> , or seed, which are managed server-side by Google. The output is therefore non-deterministic.
Gemini 3 Pro-Preview Python API	Default API settings were used: <code>temperature = 1.0</code> (recommended fixed value for Gemini 3), <code>max_output_tokens = 65,536</code> , <code>top_p = 0.95</code> , and <code>top_k = 64</code> . Dynamic thinking was enabled with <code>thinking_level = high</code> (default; thinking cannot be disabled for this model). The random seed was not set; therefore, the output is non-deterministic.

References

- [1] W.M.P. van der Aalst, *Process Mining - Data Science in Action*, Second Edition, Springer, 2016. <https://doi.org/10.1007/978-3-662-49851-4>
- [2] M. Reichert, B. Weber, *Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies*, Springer, 2012. <https://doi.org/10.1007/978-3-642-30409-5>
- [3] J. Desel, W. Reisig, G. Rozenberg, (Eds.), *Lectures on Concurrency and Petri Nets*, Advances in Petri Nets [This tutorial volume originates from the 4th Advanced Course on Petri Nets, ACPN 2003, held in Eichstätt, Germany in September 2003. In addition to lectures given at ACPN 2003, additional chapters have been commissioned.] Vol. 3098 of Lecture Notes in Computer Science, Springer, 2004. <https://doi.org/10.1007/b98282>
- [4] P.A. de Clercq, K. Kaiser, A. Hasman, Computer-interpretable guideline formalisms, in: A. ten Teije, S. Miksch, P.J. F. Lucas, (Eds.), *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*, Vol. 139 of Studies in Health Technology and Informatics, IOS Press, 2008, pp. 22–43. <https://doi.org/10.3233/978-1-58603-873-1-22>
- [5] B.A.R. Blog, The shift from models to compound AI systems, 2024, Berkeley AI Research Blog, <https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/>.
- [6] A. Korotich, Cpo predictions: The year ai finally learns to speak workflow, 2025, <https://www.wrike.com/blog/ai-workflow-2026/>. Industry analysis emphasizing cohesive AI ecosystems over standalone tools.
- [7] S. Palani, G. Ramos, Orchid: Orchestrating context across creative workflows with generative AI, 2025. [arXiv:2508.19517](https://arxiv.org/abs/2508.19517)
- [8] Italian Stroke Association, Linee guida attuali, 2023, <https://isa-aii.com/linee-guida/linee-guida-attuali/>.
- [9] S. Dunzer, M. Stierle, M. Matzner, S. Baier, Conformance checking: a state-of-the-art literature review, in: S. Betz (Eds.), *Proceedings of the 11th International Conference on Subject-Oriented Business Process Management, S-BPM ONE 2019*, ACM, Seville, Spain, 2019, pp. 4:1–4:10. <https://doi.org/10.1145/3329007.3329014>
- [10] A. Rozinat, W.M.P. van der Aalst, Conformance checking of processes based on monitoring real behavior, *Inf. Syst.* 33 (2008) 64–95. <https://doi.org/10.1016/j.is.2007.07.001>
- [11] S.J.J. Leemans, D. Fahland, W.M.P. van der Aalst, Scalable process discovery and conformance checking, 2018, pp. 599–631. <https://doi.org/10.1007/s10270-016-0545-x>
- [12] A. Adriansyah, J. Munoz-Gama, J. Carmona, B. van Dongen, W. van der Aalst, Alignment based precision checking, in: M.L. Rosa, P. Soffer(Eds.), *Business Process Management Workshops - BPM 2012 International Workshops*, Tallinn, Estonia, September 3, 2012. Revised Papers, Springer, 2012, pp. 137–149. https://doi.org/10.1007/978-3-642-36285-9_15
- [13] D. Borrego, I. Barba, Conformance checking and diagnosis for declarative business process models in data-aware scenarios, *Expert Syst. Appl.* 41 (2014) 5340–5352. <https://doi.org/10.1016/J.ESWA.2014.03.010>
- [14] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [15] G. DeepMind, Gemini 2.5 flash, 2025, [https://en.wikipedia.org/wiki/Gemini_\(language_model\)](https://en.wikipedia.org/wiki/Gemini_(language_model)). Large language model developed by Google, optimized for low-latency and efficient multimodal reasoning. part of the Gemini family of AI models; designed for fast, cost-effective AI reasoning and multimodal tasks.
- [16] D. Khurana, A. Koli, K. Khatter, S. Singh, Natural language processing: state of the art, current trends and challenges, *Multim. Tool. Appl.* 82 (2023) 3713–3744. <https://doi.org/10.1007/s11042-022-13428-4>
- [17] A. Berti, H. Kourani, H. Hafke, C. Li, D. Schuster, Evaluating large language models in process mining: capabilities, benchmarks, and evaluation strategies 2024. [arXiv:2403.06749](https://arxiv.org/abs/2403.06749)
- [18] A. Berti, D. Schuster, W.M.P. van der Aalst, Abstractions, scenarios, and prompt definitions for process mining with llms: a case study, in: J.D. Weerd, L. Pufahl (Eds.), *Business Process Management Workshops - BPM 2023 International Workshops*, Utrecht, The Netherlands, Springer, Revised Selected Papers, 2023, pp. 427–439. https://doi.org/10.1007/978-3-031-50974-2_32
- [19] H. Kourani, A. Berti, J. Hennrich, W. Kratsch, R. Weidlich, C. Li, A. Arslan, D. Schuster, W.M.P. van der Aalst, Leveraging large language models for enhanced process model comprehension, 2024. [arXiv:2408.08892](https://arxiv.org/abs/2408.08892)
- [20] M. Qafari, W. van der Aalst, Fairness-aware process mining, in: H. Panetto, C. Debruyne, M. Hepp, D. Lewis, C.A. Ardagna, R. Meersman (Eds.), *On the Move to Meaningful Internet Systems: OTM 2019 Conferences*, Springer International Publishing, Cham, 2019, pp. 182–192.
- [21] H. Kourani, A. Berti, D. Schuster, W. van der Aalst, PromoAI: process modeling with generative AI, in: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence Organization, 2024b.
- [22] A. Susaiyah, N. Sidorova, Zero-Shot Approaches for Extraction of Event Logs from Medical Notes, *Springer Nature Switzerland*, Cham, pp. 435–451.
- [23] M. Grohs, L. Abb, N. Elsayed, J. Rehse, Large language models can accomplish business process management tasks, 2023. [arXiv:2307.09923](https://arxiv.org/abs/2307.09923)
- [24] U. Jessen, M. Sroka, D. Fahland, Chat-chat or deep talk: Prompt engineering for process mining, 2023. [arXiv:2307.09909](https://arxiv.org/abs/2307.09909)
- [25] H. Yang, M. Li, H. Zhou, Y. Xiao, et al., Large language model synergy for ensemble learning in medical question answering: design and evaluation study, *J. Med. Internet Res.* (2025). <https://pubmed.ncbi.nlm.nih.gov/40658884>.
- [26] C. Cosentino, A. Defilippo, M. Dossena, C. Irwin, S. Joubbi, P. Liò, Healthbranches: synthesizing clinically-grounded question answering datasets via decision pathways, 2025. [arXiv:2508.07308](https://arxiv.org/abs/2508.07308)
- [27] G. Rodella, A. Scalogna, L. Carenzo, F. Della Corte, From prompt to platform: an agentic ai workflow for healthcare simulation scenario design, *Adv. Simul.* 10 (2025). <https://doi.org/10.1186/s41077-025-00357-z>
- [28] Google Labs, NotebookLM, 2025, AI-powered research and note-taking tool by Google that synthesizes documents using Gemini models. <https://en.wikipedia.org/wiki/NotebookLM>. uses Gemini family large language models (e.g., Gemini 2.5 Flash, Gemini 3) for document understanding and synthesis.
- [29] Google DeepMind, Gemini 3 Pro, advanced large language model by Google with high reasoning and multimodal capabilities, 2025b, [https://en.wikipedia.org/wiki/Gemini_\(language_model\)](https://en.wikipedia.org/wiki/Gemini_(language_model)). Part of the Gemini 3 series, combining deep reasoning, multimodal processing, and enhanced performance.
- [30] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing, *ACM Comput. Surv.* 55 (2023) 1–35.
- [31] P. Sahoo, A. Singh, S. Saha, V. Jain, S. Mondal, A. Chadha, A systematic survey of prompt engineering in large language models: techniques and applications, 2025. [arXiv:2402.07927](https://arxiv.org/abs/2402.07927)
- [32] A. Bottrighi, A. Canessa, D. Ferrandi, G. Leonardi, A. Maconi, C. Massarino, S. Montani, A. Roveta, M. Striani, From discharge letters to process traces with LLMs: a human-in-the-loop pipeline, in: M. Giacomini, J. Delgado (Eds.), *Proceedings of Medical Informatics Europe (MIE)*, IOS Press, 2026.
- [33] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 9459–9474.
- [34] G. Leonardi, S. Montani, M. Striani, Llm-powered conformance checking for medical processes, in: P. Bruno, F. Calimeri, F. Cauteruccio, M. Dragoni, F. Stella, G. Terracina (Eds.), *Proceedings of International Joint Workshop of Artificial Intelligence for Healthcare (HC@AIxIA) and Hybrid Models for Coupling Deductive and Inductive Reasoning (HYDRA)*, CEUR, 2025.
- [35] T. Shnitzer, A. Ou, M. Silva, K. Soule, Y. Sun, J. Solomon, N. Thompson, M. Yurochkin, Large language model routing with benchmark datasets, in: *Proceedings of the First Conference on Language Modeling (COLM)*, 2024.
- [36] L. Yang, S. Xu, A. Sellergren, T. Kohlberger, Y. Zhou, I. Ktena, A. Kiraly, F. Ahmed, F. Hormozdiari, T. Jaroensri, E. Wang, E. Wulczyn, F. Jamil, T. Guidroz, C. Lau, S. Qiao, Y. Liu, A. Goel, K. Park, A. Agharwal, N. George, Y. Wang, R. Tanno, D.G.T. Barrett, W. Weng, S. Mahdavi, K. Saab, T. Tu, S. Kalidindi, M. Etemadi, J. Cuadros, G. Sorensen, Y. Matias, K. Chou, G. Corrado, J. Barral, S. Shetty, D. Fleet, S.M.A. Eslami, D. Tse, S. Prabhakara, C. McLean, D. Steiner, R. Pilgrim, C. Kelly, S. Azizi, D. Golden, Advancing multimodal medical capabilities of Gemini, 2024. [arXiv:2405.03162](https://arxiv.org/abs/2405.03162).
- [37] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, J. Guo, A survey on LLM-as-a-judge, 2025. [arXiv:2411.15594](https://arxiv.org/abs/2411.15594)