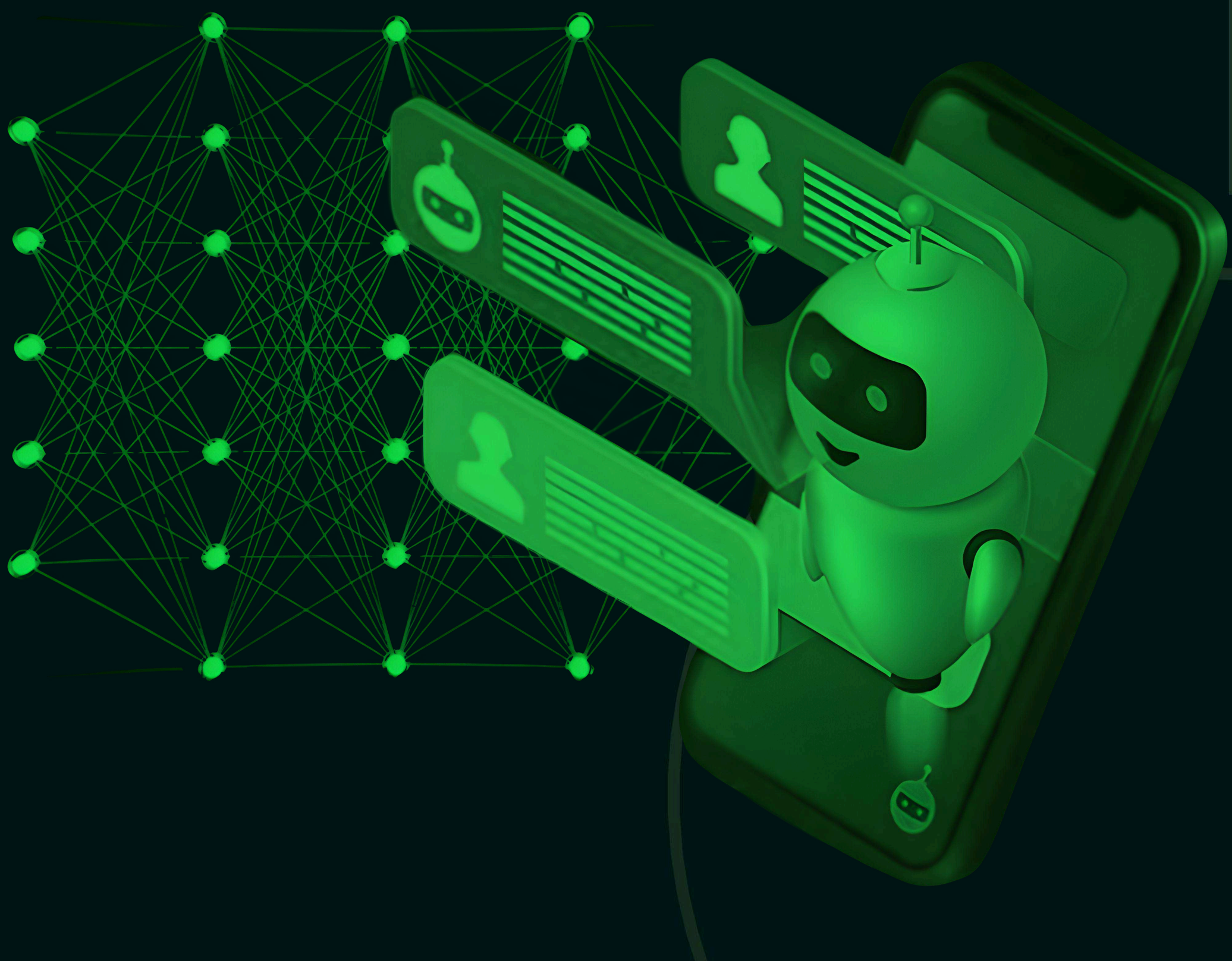
INTER IIT TECH MEET 14.0

MID PREP



OBSERVE·AI

Causal Rationale Extraction and Synthesis from Conversational Data on Business Events

INTER IIT TECH MEET 14.0

About Us

Observe.AI is the leading AI agent platform for customer experience. It enables enterprises to deploy AI agents that automate customer interactions, delivering natural conversations for customers with predictable outcomes for the business. Observe.AI combines advanced speech understanding, workflow automation, and enterprise-grade governance to execute end-to-end workflows with AI agents. It also enables teams to guide and augment human agents with AI copilots, and analyze 100% of human and AI interactions for insights, coaching, and quality management. Companies like DoorDash, Affordable Care, Signify Health, and Verida use Observe.AI to transform customer experiences every day by accelerating service speed, increasing operational efficiency, and strengthening customer loyalty across every channel. Backed by Menlo Ventures, Next47, NGP Capital, Scale Ventures, Nexus Ventures, and Y-Combinator, Observe.AI has its office in San Francisco, US and Bangalore, India.

Problem Statement Description

In large-scale customer-interaction operations, contact-centres routinely process tens thousands to hundreds of thousands of agent-customer dialogues. Among these, many conversations culminate in business-critical outcomes, such as customer escalations to supervisors, refund requests, or signals of churn. These events are not isolated or trivial: they carry cost, risk, operational overhead and may undermine service quality or brand reputation. Critically, the triggers of these outcomes are rarely singular or obvious; rather, they emerge from patterns of conversational behaviour, agent responses,

customer hesitations, branching dialogue flows, repeated queries, silences, and mis-understandings. Currently, many monitoring systems flag that an adverse event occurred (for instance “escalation on call #123”) but provide little insight into why it happened: which turns or segments of the dialogue triggered the escalation; which conversational cues systematically lead to refunds; what temporal patterns presage churn-intent.Without this visibility, organisations cannot systematically perform root-cause analysis, coach agents precisely, redesign processes or intervene proactively across the corpus. A robust technical solution must ingest large volumes of transcript data (with speaker-labels, turn-indexing, optionally timings), model conversational dynamics, map dialogue flows to business events, and surface the specific dialogue spans (utterances/turns) that most likely causally contributed to the event. Even further, such a system should enable analytic querying across the call-corpus, e.g.,“what conversational patterns lead to escalations in billing discussions?”, to identify recurring causal motifs, cluster them and provide summary insights. Operationally, the stakes are high: reducing escalations lowers cost, improves customer experience, protects brand risk and enables workforce efficiency. Technically, the landscape is challenging: transcripts are noisy (especially if derived from ASR), speaker-roles and turn boundaries may be imperfect, the event-labels are sparse, conversation lengths vary widely, branching dialogue structures complicate detection of causal spans, and retrieval over large call corpora must scale. Moreover, justification of identified spans (so that human analysts or coaches accept recommendations) adds an interpretability requirement. The solution must therefore combine detection, span-extraction, retrieval, ranking and explanation modules at scale. In sum, being able to pinpoint the conversational triggers of business-events transforms the monitoring function from mere outcome-reporting to causal insight-driven intervention.

INTER IIT TECH MEET 14.0

Tasks

1. **<u>Task 1 - Query-Driven Evidence Based Causal Explanation</u>**
    a. The objective of this task is to design a system that can accept a natural-language query related to a specific business event, such as “Why are escalations happening on calls?”, and generate an evidence-based causal explanation. The system should be capable of processing large-scale conversational transcripts to infer causal relationships between dialogue dynamics and the specified event. It must identify the key contributing factors, behaviours, and dialogue spans that serve as supporting evidence, demonstrating how particular conversational elements correlate with or lead to the event of interest. The outcome should be a structured, data-driven explanation that clearly articulates the underlying causal mechanisms connecting conversational patterns to business outcomes, providing interpretable and actionable insights rather than simple correlations.

    b. Expected Outcomes:

    i. ML System for generating a response for the given query with evidential references over the corpus of data
    ii. Dataset of simulated/real user-queries and corresponding outputs (response) from the proposed systems
    iii. Choice of evaluation metrics to evaluate the system components and/or response quality

2. **<u>Task 2 - Conversational Follow-Up and Contextual Response Generation</u>**
    a. Building upon the system developed in Task 1, this task focuses on extending its functionality to support natural conversational interaction and iterative analytical dialogue. The enhanced system should be able to handle

follow-up questions that are contextually linked to prior analyses, maintaining awareness of the user's previous queries and responses.

b. Expected Outcomes:

i. ML System for generating a response for the given query with evidential references over the corpus of data
ii. Dataset of simulated/real user-queries and corresponding outputs (response) from the proposed systems
iii. Choice of evaluation metrics to evaluate the system components and/or response quality

Dataset

A sample dataset will be provided along with the problem statement document, and the actual dataset will be released on 15th November 2025.

Deliverables and Submission

1. **<u>System Implementation</u>**

a. Provide a complete end-to-end dockerized implementation of the system covering both **Task 1** and **Task 2**.
b. Ensure the codebase is well-structured, documented, and reproducible.
c. **Do not upload** the implementation to any publicly accessible repositories (e.g., GitHub)

d. **General Guidance:**

i. **README File:** Your submission must include a comprehensive README.md file. This file should provide clear, step-by-step instructions on how to set up the environment, build the Docker image, and run the system (including any data preprocessing scripts, model training, and the final application).

ii. **Requirements:** You must provide a requirements.txt file (or an equivalent like environment.yml for Conda) that lists all project dependencies with their specific versions (e.g., pandas==2.1.0, torch==2.0.1). This is critical for ensuring your solution is reproducible.

iii. **Language:** Python is the preferred language for implementation. However, you are free to use other languages, and there will be no penalty for doing so, provided the solution is fully dockerized and reproducible as per the guidelines.

2. Technical Report

a. Submit a detailed report outlining the overall approach, system design, and methodology for both tasks.

b. Include comprehensive descriptions of data processing, model/system architecture, inference strategies, and algorithmic choices.

c. Present quantitative and qualitative evaluation results, with clear performance metrics, baseline comparisons, ablation studies, and interpretability analysis.

d. Conclude with discussion on system limitations, and directions for potential improvement.

e. **General Guidance:**

i. **Recommended Length:** A well-rounded report is expected to be in the **6-8** page range.

ii. **Formatting Tool:** Using **LaTeX** is the recommended tool for writing and formatting your report.

iii. **Content Depth:** There is no strict restriction on report length. However, we anticipate that reports with just 4-5 pages may not contain sufficient detail, results, and elaboration to be considered comprehensive

3. Simulated/Real Query–Sytem Output Submission

a. Provide a curated set of **simulated/real user queries** relevant to key business events (e.g., escalations, refunds, churn, product improvements).
b. Include the corresponding **system outputs** demonstrating evidence-based causal explanations and context-aware conversational follow-ups.
c. Describe the **simulation framework** within the technical report that is used for dataset generation or testing, emphasizing how simulation was applied to evaluate and refine system performance
d. **General Guidance:**

A good submission must include a curated dataset of minimum 50-100 queries, all either simulated or a combination of real and simulated queries. This dataset is crucial for demonstrating your system's capabilities and evaluation process.

i. **Query Categorization:** These queries must be organized into distinct, meaningful categories. An ideal submission will define **functionally relevant categories** that rigorously test the system's analytical power. Such as category of queries by -> **By Task:** Task 1 (Initial Causal Inquiry) vs. Task 2 (Contextual Follow-up); **By Difficulty:** Simple fact retrieval, complex multi-hop reasoning, or queries that test ambiguity; **By Use-Case:** Understanding agent's behavior vs. Finding information about product feedback etc.
    - **Distribution:** Ensure a reasonable and proportionate number of queries across your defined categories.

ii. **Query Simulation Process:** To efficiently generate and refine this query dataset, we recommend you explore modern curation techniques. Participants are encouraged to combine automated and human-guided methods to create and refine their query dataset. Use **LLMs** to generate realistic queries (e.g., from an Agent Coach perspective), refine them through **Human-in-the-Loop (HITL)** review for quality and relevance, and optionally apply an **LLM-as-Judge** framework to automatically assess query diversity, realism, and contextual fit.

- Consider how this framework is generalizable such that it can be applied efficiently and effectively across domains, e.g., from airline reservations to banking to insurance, so it remains robust even as the underlying dataset and query types change.

iii. **Format of CSV:** Should contain 4 columns - Query_Id, Query, Query_Category, System Output, Remarks. You may additionally add any further columns as deemed necessary for conveying your findings, such as multiple approaches output, multi-level query categorization etc.

4. Presentation

a. Prepare a concise presentation summarising the objectives, methodology, system architecture, results, and key insights from Tasks 1 and 2.

b. A portion of the presentation should show a live or recorded demonstration of the system that is built.

c. General Guidance:

A strong presentation should be concise, well-timed, and focused on clarity rather than length. The key is to clearly communicate the overall approach, system design, key results, and demonstrate the system in action. Avoid covering every technical detail, those should be included in the technical report. Instead, focus on presenting the core insights, main takeaways, and how the audience can explore further details in the accompanying documentation.

1st, 2nd and 3rd deliverables should be submitted within a single zip file.

INTER IIT TECH MEET 14.0

Overall Evaluation Philosophy

The evaluation of submission does not focus exclusively on the accuracy or absolute quality of the system's responses in Task 1 and Task 2. Instead, emphasis is placed on the rigor and depth of the overall process, how participants design, benchmark, simulate, iterate, and reason about their systems. Strong submissions will demonstrate a clear developmental journey: outlining the starting point, the rationale behind design choices, what approaches worked or failed, how iterations evolved, what insights were gained, and what further experiments might be pursued with more time or resources. The assessment therefore values thoughtful experimentation, systematic reflection, and clarity of explanation as much as technical achievement.

1. **Presentation (20%)**
   a. Structured flow and clarity in communication of results and takeaways (10%)
   b. Live or Recorded Demo (5%)
   c. Question and Answer (5%)

2. **Approach and Architecture (25%)**
   a. Clarity, soundness, and innovation of the overall ML system and architecture. (10%)
   b. Explanation of modelling choices, including rationale behind key design decisions and exploration of alternative approaches. (15%)

3. **Dataset of simulated/real queries and outputs from the proposed system (25%)**
   a. Submission of Dataset (Deliverable #3) – Dataset includes at least two kinds of question–answer pairs: questions that the proposed system can handle effectively, and questions it cannot respond to appropriately, categorized via chosen evaluation metrics or manual annotation. Along with the queries, corresponding system outputs for both kinds of question-answer pairs need to be submitted. (10%)

b. Clarity in explaining the simulation framework or process for generating synthetic and/or real user queries related to business events, along with categories of queries. (10%)

c. Demonstration of how outputs from simulated queries were utilised to iteratively evaluate, refine, and improve the overall system performance. (5%)

4. Evaluation of System for Task 1 and Task 2 (30%)

a. Choice of Evaluation Metrics and Methods (10%) – Appropriateness of metrics used to measure response quality, causal explanation quality / evidence provided for the explanation, and conversational coherence.
b. Comparison of Alternatives and Baselines (10%) – Rigor in benchmarking the proposed approach against existing or alternative methods, including discussion of relative strengths and weaknesses.
c. Ablation Studies and/or Error Analysis (10%) – Depth of diagnostic experiments analysing component contributions, error cases, and system limitations, supported by objective results/analysis

INTER IIT TECH MEET 14.0

Additional Notes

- The goal of this problem statement is not merely to optimize for specific evaluation metrics. Simply designing easy queries to achieve perfect results is not the ideal approach. Instead, participants are encouraged to demonstrate how their system performs across diverse query types, highlighting both its strengths and limitations, and to explore ways to iterate or enhance the system to address those challenges. This is how the evaluation scoring is also designed.
- Participants are free to use their choice of Large Language Models (LLMs), open-source or proprietary, large or small, including OpenAI, Anthropic, Llama, Qwen, DeepSeek, Amazon Nova, and others. The evaluation does not score usage of any particular LLM favorable or otherwise. For comparative studies or ablations, exploration can be limited to a maximum of three configurations or approaches to maintain focus on addressing the tasks in the given time window.
- It is recommended to use LLMs and related components, such as embeddings, rerankers, or other out-of-the-box (OOB) tools, rather than undertaking extensive custom training, given the time and effort constraints. However, if participants can justify the need for custom training by clearly demonstrating the limitations of OOB artifacts and how their tailored approach addresses those gaps, it will be considered a valuable and commendable contribution.
- Participants are free to directly adopt and adapt any existing end-to-end systems that may already exist, as part of their solution. While there is no penalty as such for such adoptions and adaptations, participants must justify the choice and showcase rigorous benchmarking with any alternative approaches.

INTER IIT TECH MEET 14.0

IP Rights Disclaimer

All intellectual property (IP) rights, including but not limited to copyrights, patents, and trade secrets, for any solutions, code, designs, or derived works created in response to the Observe.AI Problem Statement shall remain the exclusive property of Observe.AI.

By participating and submitting a solution, all team members acknowledge and agree that:

1. Observe.AI shall retain full ownership of the submitted solution and any derivative works arising from it.
2. Participants shall not reuse the submitted codebase, proprietary data, or technical documentation outside the scope of this competition.

However, Observe.AI grants participants limited rights for professional recognition, as outlined below:

1. Professional Announcements (e.g., LinkedIn): Participants are permitted to share general, non-technical statements about their participation or achievement. They must not disclose proprietary data, share any portion of the solution code, or reveal technical or architectural details of the project.
2. Codebase and Documentation: The complete codebase, proprietary datasets, and any detailed technical documentation are strictly excluded from public sharing, posting, or reuse.
3. Resume/CV Mentions: Participants may list their participation and achievements in their resumes or CVs. The description must remain generic and high-level, without including confidential details, dataset specifics, or performance metrics related to Observe.AI's business.

By continuing participation in this Problem Statement, all team members agree to adhere to the above terms.