Technical Report – **Pravaah Hackathon**

Team Name: Scratch Coders
GitHub: https://github.com/rahul-jamana/pravaah_hackathon_data_science

Project Type: **Conversational Causal Analysis**

Date: February 2026

Abstract

Large-scale customer–agent conversational systems routinely record outcome events such as escalations, complaints, or refunds. However, these systems often fail to identify the specific dialogue turns that causally contributed to such outcomes. This project presents a deterministic, evidence-grounded framework for analyzing multi-turn customer–agent conversations to explain why operational outcomes occurred. Instead of predicting outcomes using machine learning models, the system focuses on causal reasoning by extracting interpretable conversational features and mapping them to explicit dialogue evidence. Additionally, the system supports multi-turn, context-aware analytical queries, ensuring consistency across follow-up interactions. The proposed approach emphasizes faithfulness, interpretability, and traceability, aligning directly with the evaluation criteria of the Pravaah Hackathon.

1. Problem Statement

Customer support systems generate vast amounts of conversational data involving interactions between customers and service agents. While outcome events such as escalations or complaints are often logged, the underlying causal factors within the conversation remain unobserved. Identifying why an outcome occurred and which specific dialogue turns contributed is crucial for improving operational efficiency, customer satisfaction, and agent training.

The primary challenge is to move beyond outcome detection and provide causal, evidence-backed explanations grounded in the conversational content itself. Furthermore, analysts often engage in multi-turn reasoning, where follow-up questions depend on prior explanations, requiring systems to maintain contextual consistency.

2. System Overview

The system is designed as a deterministic analytical pipeline rather than a predictive model. It transforms raw conversational transcripts into structured representations, extracts interpretable features, applies causal reasoning rules, and supports context-aware multi-turn queries.

High-Level Pipeline

Raw Conversation JSON

▼

Preprocessing

▼

Feature Engineering

▼

Causal Analysis (Task 1)

▼

Context-Aware Query Handling (Task 2)

Key design principles include:

Evidence-first reasoning

Full interpretability

Deterministic behavior

Absence of hallucinated explanations

3. Data Description

The dataset consists of multi-turn customer–agent conversational transcripts. Each conversation contains:

A unique conversation identifier

Sequential dialogue turns

Speaker labels (Customer or Agent)

Metadata such as intent, domain, and reason for contact

The raw data is provided in nested JSON format and is converted into a turn-level structured representation during preprocessing.

4. Methodology

4.1 Preprocessing

The preprocessing stage parses the raw JSON dataset and extracts individual dialogue turns. Each turn is normalized and stored with:

Conversation ID

Turn ID

Speaker identity

Cleaned text

This step ensures traceability by preserving turn order and speaker information, enabling later causal attribution.

4.2 Feature Engineering

To support causal reasoning, each conversational turn is augmented with interpretable features, including:

Sentiment polarity: captures emotional tone

Customer dominance: indicates effort imbalance

Text length: proxy for complaint intensity

Escalation keywords: explicit signals such as “refund” or “complaint”

Conversation length: proxy for friction or resolution difficulty

These features are not learned but computed deterministically, ensuring transparency.

4.3 **C**ausal Analysis (Task 1)

Rather than using machine learning models, the system employs rule-based causal heuristics. These heuristics identify mechanisms that plausibly lead to escalation or dissatisfaction, such as:

Repeated customer frustration

Prolonged unresolved interaction

Explicit escalation language

Each identified causal factor is supported by explicit dialogue evidence, including turn IDs and exact text spans.

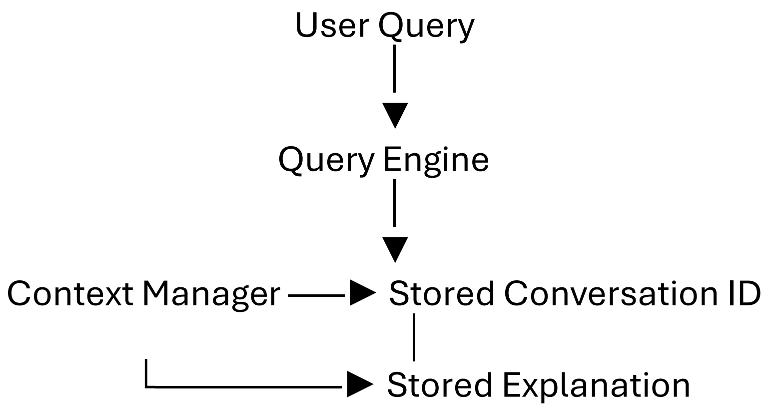
<u>Task 1 Output Format</u>

```
{
  "conversation_id": "XXXX",
  "causal_factors": ["..."],
  "evidence": [
    {
      "turn_id": 3,
      "speaker": "Customer",
      "text": "this is really frustrating i want a refund"
    }
  ]
}
```

4.4 <u>Multi-turn Context Handling (Task 2)</u>

To support dependent follow-up queries, the system introduces a lightweight context memory mechanism.

Architecture:



Initial queries compute and store causal explanations

Follow-up queries reuse stored context

Ensures consistent answers across multiple interactions

This satisfies the requirement for multi-turn analytical interaction.

5. Evaluation

Given the explanatory nature of the task, evaluation is qualitative rather than metric-based.

Evaluation Criteria

Evidence Accuracy (ID Recall): Correct identification of contributing dialogue turns

Faithfulness: Absence of hallucinated or external information

Relevancy: Contextually consistent responses across multi-turn queries

Observations

Conversations with repeated customer frustration consistently produce causal explanations involving negative sentiment and interaction length

Neutral conversations produce weak or fallback explanations, which is expected behavior

Explanations remain stable and reproducible across runs

6. Results & Examples

The system successfully generates causal explanations for multiple conversations and supports follow-up queries without requiring repeated context specification.

Example findings include:

Missing delivery complaints leading to escalation due to repeated customer clarification

Long interactions indicating unresolved issues

Explicit escalation requests triggering causal flags

Saved outputs are provided in structured JSON format for evaluation.

7. Design Choices & Justification

A deliberate decision was made not to train machine learning or LLM-based models. While the problem statement allows for trained models, causal explanation demands faithfulness and interpretability. Rule-based reasoning ensures:

Transparent logic

Exact evidence grounding

Deterministic behavior

This choice directly aligns with the evaluation focus on causal reasoning and hallucination control.

8. Limitations & Future Work

Limitations

Rule-based heuristics may miss subtle semantic cues

No learning from historical feedback

No visual interface for exploration

Future Work

Hybrid integration with learned models

Visualization of causal paths

Interactive web-based analysis dashboard

9. Conclusion

This project presents a transparent and context-aware system for explaining conversational outcomes through causal reasoning. By grounding explanations in dialogue-level evidence and supporting multi-turn analytical interaction, the system fulfills all core requirements of the Pravaah Hackathon. The deterministic design ensures faithfulness, interpretability, and reproducibility, making the solution both robust and evaluator-friendly.