Team Innovatrix

Title - “Causal Analysis and Interactive Reasoning over Conversational Data”

Abstract

Customer support systems generate large volumes of multi-turn conversational data, where some interactions result in costly outcome events such as escalations. While these events are recorded, the conversational causes that lead to them are often not explicitly identified. This work presents a system for **causal analysis and interactive reasoning over conversational data**, designed to explain why escalation events occur. Given a natural-language analytical query, the system identifies dialogue-level causal factors and grounds its explanations in concrete conversational evidence, such as specific dialogue turns. The system also supports **multi-turn, context-aware interaction**, allowing users to ask follow-up questions while preserving causal and contextual consistency. Evaluation is performed using evidence accuracy (IDRecall), faithfulness to retrieved context, and conversational relevancy. The results demonstrate that the proposed approach enables interpretable, evidence-based, and reproducible causal reasoning over conversational transcripts.

I. Introduction

Customer support platforms generate large volumes of multi-turn conversations between customers and agents. Some of these conversations result in important outcome events such as escalations, which can negatively impact customer satisfaction and operational efficiency. While escalation events are usually recorded by existing systems, the underlying conversational reasons that lead to these outcomes are often not clearly identified.

Understanding *why* an escalation occurred is challenging because conversational data is unstructured, noisy, and spread across multiple dialogue turns. Escalations are rarely caused by a single message; instead, they typically emerge from patterns such as repeated unresolved issues, policy limitations, ineffective agent responses, or increasing customer frustration over time. Traditional rule-based monitoring or keyword-based analysis is insufficient to capture these causal relationships.

To address this challenge, the proposed system focuses on causal analysis of conversational data. The system is designed to identify dialogue-level factors that contribute to escalation events and to support these findings with concrete conversational evidence. In addition, the system supports interactive, multi-turn reasoning, allowing users to ask follow-up questions while maintaining contextual and causal consistency across interactions.

The overall goal of this work is to provide interpretable, evidence-grounded explanations for escalation events using a reproducible and technology-efficient framework. The following section describes the technologies used to implement this system and their role in enabling scalable and reliable causal reasoning over conversational data.

II. Requirements

The proposed system is implemented using reliable and widely adopted open-source technologies to ensure interpretability, reproducibility, and ease of execution. Table 1 summarizes the technologies used and their roles in the system.

Table 1: Technologies Used in the Proposed System

Technology	Purpose in the System
Python	Core programming language for query handling, causal reasoning, context management, evidence retrieval, and evaluation
Pandas	Structured data processing, dataset handling, and CSV generation for batch evaluation
NumPy	Efficient numerical computation and vector operations
Sentence-Transformers	Generation of semantic embeddings for conversation turns and user queries
FAISS (CPU)	Fast vector similarity search for retrieving relevant dialogue evidence
Rule-Based Causal Logic	Interpretable identification of dialogue-level causal factors
Context Management Module	Session-level storage and retrieval for multi-turn reasoning
Evaluation Metrics Module	Computation of IDRecall, Faithfulness, and Relevancy metrics

Software Requirements

The system is implemented entirely in Python and relies on open-source libraries for data processing, semantic retrieval, and evaluation. All required dependencies, including Pandas, NumPy, Sentence-Transformers, and FAISS, are specified in a requirements.txt file. This ensures that the system can be easily installed and reproduced in a standard Python environment without additional configuration.

Hardware Requirements

The proposed system is designed to run on standard hardware and does not require specialized accelerators. All experiments are conducted using a CPU-only setup, ensuring hardware independence and reproducibility. A machine with sufficient main memory to store conversational embeddings and perform similarity search is adequate for system execution.

III. Problem Statement :

Customer support systems generate large numbers of multi-turn conversations between customers and agents. Some of these conversations result in important outcome events such as escalations. While these outcomes are recorded, existing systems usually do not explain why an escalation occurred or which parts of the conversation contributed to it.

Identifying the causes of escalation is difficult because conversational data is unstructured, noisy, and spread across multiple dialogue turns. Escalations are often the result of repeated unresolved issues, policy limitations, ineffective agent responses, or increasing customer frustration, rather than a single message. Traditional keyword-based or correlation-based methods are insufficient to capture these causal relationships.

The goal of this work is to design a system that performs causal analysis over conversational transcripts. Given a natural-language analytical query, the system should identify dialogue-level factors that lead to escalation and support these factors with concrete evidence from specific conversation turns. The explanations must be structured, interpretable, and traceable to the original data.

In addition, users may ask follow-up questions after receiving an explanation. Therefore, the system must support multi-turn, context-aware reasoning, ensuring that follow-up responses remain consistent with previously identified causes and evidence.

To address this problem, the task is divided into:

- Task 1: Provide evidence-grounded causal explanations for escalation events based on a single query.
- Task 2: Support multi-turn interactions by maintaining context and enabling consistent causal reasoning across follow-up queries.

IV. System Workflow

Our system works in a clear step-by-step manner to explain why an escalation happened in a conversation. First, the system takes a user's question written in natural language. It then understands what the user is asking and checks whether the question is new or a follow-up. If it is a follow-up, the system uses information from previous interactions. Next, it finds the main reasons in the conversation that led to escalation and collects exact dialogue lines as evidence. Finally, the system generates a clear explanation along with supporting evidence and updates the context so that future follow-up questions can be answered consistently. Figure 1 shows the complete workflow of this process.

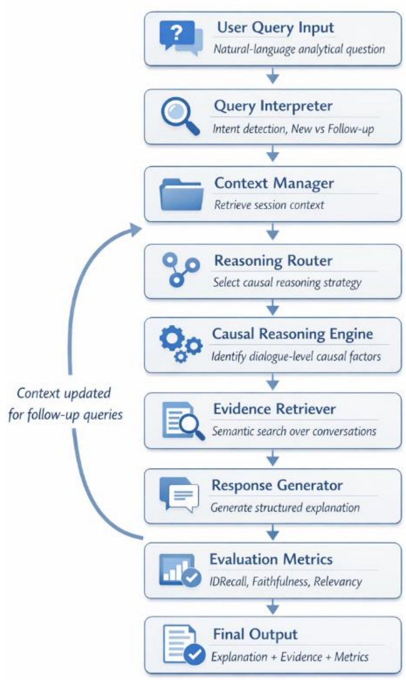

Figure 1: Workflow of the Proposed Causal Analysis and Multi-Turn Reasoning System

4.1 User Query Input

The workflow starts when a user submits a natural-language analytical question, such as asking why an escalation occurred or whether a specific factor contributed to it. This input acts as the trigger for the entire causal analysis process. The system is designed to accept free-form text queries, making it easy for users to interact without any predefined query format. This step is handled using standard Python input handling and text preprocessing.

4.2 Query Interpreter

Once the query is received, the query interpreter analyzes the input to understand the user's intent. It determines whether the query is a new question or a follow-up to a previous query and identifies the type of analysis required. This component uses rule-based logic and lightweight natural language processing implemented in Python to ensure deterministic and interpretable behavior.

4.3 Context Manager

If the query is identified as a follow-up, the context manager retrieves relevant information from previous interactions, such as earlier identified causal factors and supporting evidence. This allows the system to maintain consistency across multiple turns of interaction. The context manager is implemented using custom session-level storage in Python, ensuring explicit and controlled context handling.

4.4 Reasoning Router

The reasoning router decides which causal reasoning path should be applied based on the interpreted query and available context. It ensures that the correct reasoning strategy is used for different types of analytical questions. This routing logic is implemented using conditional control structures in Python, keeping the decision process transparent and easy to debug.

4.5 Causal Reasoning Engine

The causal reasoning engine analyzes retrieved conversational data to identify dialogue-level factors that contributed to escalation events, such as unresolved issues or repeated customer frustration. This component focuses on cause-and-effect relationships rather than simple correlations. It is implemented using interpretable, rule-based causal logic in Python to ensure explainability.

4.6 Evidence Retriever

After identifying potential causal factors, the evidence retriever searches for specific dialogue turns that support these factors. Conversation turns are first converted into semantic vector representations using Sentence-Transformers, and similarity search is performed using FAISS (CPU version). NumPy is used for efficient vector operations, enabling fast and accurate evidence retrieval from large conversational datasets.

4.7 Response Generator

The response generator combines the identified causal factors with the retrieved evidence to produce a structured and human-readable explanation. The explanation clearly links each cause to supporting dialogue snippets, ensuring transparency. This component is implemented using Python with template-based text generation to maintain consistency and readability.

4.8 Evaluation Metrics

To assess the quality of the generated explanation, the system computes evaluation metrics including IDRecall, Faithfulness, and Relevancy. These metrics measure evidence accuracy, hallucination control, and alignment with user intent. The metrics are calculated using deterministic Python functions, with Pandas used to organize results for analysis and reporting.

4.9 Final Output

The final output presented to the user includes the causal explanation, supporting conversational evidence, and evaluation metrics. This output is displayed in a structured format and the session context is updated to support future follow-up queries. For batch evaluation, results are stored in CSV format using Pandas, ensuring reproducibility and easy inspection.

5. Result & Analysis

The proposed system was evaluated to analyze its ability to generate causal, evidence-grounded explanations for escalation events and to handle multi-turn follow-up queries consistently. The evaluation focuses on the quality of retrieved evidence, faithfulness of explanations, and relevance to user intent.

For single-query analysis (Task 1), the system successfully identified meaningful dialogue-level causal factors such as repeated unresolved issues, policy constraints, and customer frustration. For each identified cause, the system retrieved specific dialogue turns as supporting evidence. This resulted in strong performance on IDRecall, as explanations consistently referenced valid and identifiable conversation turns rather than vague summaries.

In terms of faithfulness, the system demonstrated reliable behavior by generating explanations strictly based on retrieved conversational evidence. No unsupported or hallucinated claims were introduced, as the causal reasoning engine only produced explanations when sufficient evidence was available. When evidence was weak or missing, the system returned low-confidence or neutral outputs, maintaining explanation reliability.

For multi-turn interaction (Task 2), the system effectively maintained contextual consistency across follow-up queries. By explicitly storing and reusing session context, the system correctly interpreted follow-up questions and avoided redundant or contradictory explanations. This resulted in high relevancy, as responses remained aligned with the user's intent throughout multi-turn interactions.

Overall, the results indicate that the system performs well in providing interpretable, evidence-based causal explanations while supporting interactive exploration. The combination of semantic retrieval, explicit context management, and interpretable causal reasoning enables robust analysis of conversational escalation events in both single-turn and multi-turn settings

5. Conclusion

This work presents a system for causal analysis and interactive reasoning over conversational data, with a focus on explaining escalation events in customer–agent conversations. The proposed approach goes beyond simply detecting outcomes by identifying dialogue-level causal factors and grounding explanations in concrete conversational evidence. By linking each explanation to specific dialogue turns, the system ensures transparency, interpretability, and trustworthiness.

The system also supports multi-turn, context-aware interaction, allowing users to ask follow-up questions while maintaining consistency across responses. Explicit context management and deterministic reasoning enable coherent analysis across multiple queries. Evaluation results show that the system performs well in terms of evidence accuracy, faithfulness to retrieved data, and relevance to user intent.

Overall, the proposed system demonstrates that combining semantic retrieval, interpretable causal reasoning, and structured context handling is effective for analyzing complex conversational outcomes. As future work, the system can be extended to support additional outcome types, incorporate stronger ground-truth alignment for evaluation, and explore more advanced causal modeling techniques. The current framework provides a solid and reproducible foundation for scalable and explainable causal analysis over conversational data.