A PROJECT REPORT ON

CAUSAL ANALYSIS AND INTERACTIVE REASONING OVER CONVERSATIONAL DATA

Submitted to:

Submitted by:

Subham Nayak (Team Lead)

Team HacKNomads Mohapatra S.H Gargi

B.Vineet Patro

Nikita

INSTITUTION OF TECHNICAL EDUCATION & RESEARCH

SIKSHA ‘O’ ANUSANDHAN (DEEMED TO BE UNIVERSITY)

BHUBANESWAR

1. PROBLEM DESCRIPTION

Large-scale conversational systems produce multi-turn dialogue transcripts with conversation-level outcome labels, but lack turn-level causal annotations. Existing approaches identify outcomes without explaining which dialogue segments or interaction patterns caused them. Conversational data is noisy and unstructured, requiring robust and scalable analytical methods. There is a need for interpretable systems that infer causally relevant dialogue spans and recurring patterns linked to outcomes. The system must support interactive, multi-turn analysis by maintaining contextual consistency across user queries.

The solution integrates machine learning, information retrieval, and a Flask-based backend API, supported by persistent storage using MySQL.

2. SYSTEM OBJECTIVE

The primary objectives of the system are:

2.1. Predict the most likely causal event / intent behind a user query

2.2 Retrieve supporting conversational evidence from historical transcripts

2.3 Provide explainable outputs rather than black-box predictions

2.4 Maintain conversation context across multiple user queries

2.5 Log interactions for traceability and future analysis

3. OVERALL SYSTEM ARCHITECTURE

The system consists of the following components:

3.1 Backend API (Flask):

3.1.1 Exposes REST endpoints for query analysis

3.1.2 Coordinates prediction, explanation generation, and context updates

3.2 Machine Learning Module:

3.2.1 TF-IDF vectorization of conversational text

3.2.2 Logistic Regression classifier for intent prediction

3.3 Evidence Retrieval Module

3.3.1 Filters transcripts by predicted intent

3.3.2 Extracts causal evidence using keyword-based heuristics

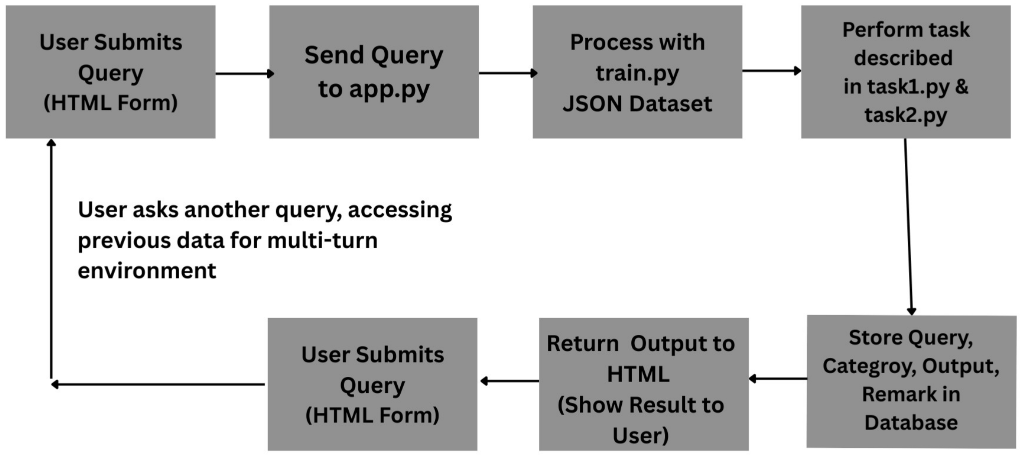
3.4 Persistence Layer (MySQL)

3.4.1 Stores all user queries, predicted events, system outputs, and remarks

3.4.2 Enables retrieval of recent interaction history

3.5 Session Memory

3.5.1 Maintains short-term conversational context during a session



4. QUERY DATASET DESCRIPTION

4.1 Dataset Source

4.1.1The system uses a JSON dataset:

4.1.2 Conversational_Transcript_Dataset.json

4.2 Dataset Structure

Each entry in the dataset contains:

4.2.1 transcript_id: Unique identifier

4.2.2 Intent: Ground-truth event label

4.2.3 Conversation: Ordered list of conversational turns

Each Conversational turn includes:

4.2.3.1 Speaker: Agent or customer

4.2.3.2 Text: Utterance content

4.3 Example Structure (Simplified)


```
{
  "transcript_id": "T001",
  "intent": "Refund_Issue",
  "conversation": [
    {"speaker": "Customer", "text": "I never received my order"},
    {"speaker": "Agent", "text": "Let me investigate the issue"}
  ]
}
```


4.4 Dataset Usage

4.4.1 Training: Full conversations concatenated into a single text document per transcript

4.4.2 Inference: Used for evidence retrieval after intent prediction

5. Model Training Methodology

5.1 Feature Engineering

5.1.1 Text from each conversation is concatenated

5.1.2 TF-IDF Vectorizer is applied

5.1.3 Maximum features: 5000

5.1.4 Captures term importance while reducing noise

5.2 Model Selection - Logistic Regression

5.2.1 Suitable for multi-class text classification

5.2.2 Interpretable coefficients

5.2.3 Efficient for moderate-sized datasets

5.3 Training Process

5.3.1 Load transcript dataset

5.3.2 Extract text and corresponding intent labels

5.3.3 Vectorize text using TF-IDF

5.3.4 Train Logistic Regression classifier

5.3.5 Persist model and vectorizer using joblib

5.4 Stored Artifacts

5.4.1 models/outcome_model.pkl

5.4.2 models/vectorizer.pkl

6. Query Processing Pipeline

When a query is submitted to the system:

6.1 Input Reception

6.1.1 Query received via /analyze endpoint (POST request)

6.2 Intent Prediction

6.2.1 Query transformed using the trained TF-IDF vectorizer

6.2.2 Logistic Regression model predicts the most probable event

6.3 Transcript Retrieval

6.3.1Dataset filtered by predicted intent

6.4 Evidence Extraction

Keyword-based scanning of conversations

Identifies turns containing causal indicators such as:

6.4.1 "never received"

6.4.2 "refund"

6.4.3 "late"

6.4.4 "wrong address"

6.4.5 "investigation"

6.5 Explanation Generation

6.5.1 Up to three supporting transcripts returned

6.5.2 Each includes relevant conversational turns

6.6 Persistence and Context Update

6.6.1 Query and results stored in MySQL

6.6.2 Session memory updated for follow-up queries

7. Evaluation Strategy and Results

7.1 Evaluation Methodology

7.1.1Since the dataset is labelled by intent, evaluation focuses on:

7.1.1.1 Correctness of intent prediction

7.1.1.2 Quality and relevance of extracted evidence

7.1.2 Evaluation performed using:

7.1.2.1 Training accuracy (during model development)

7.1.2.2 Manual inspection of explanation relevance

7.2 **Observed Results**

7.2.1 **Intent Prediction**

7.2.1.2 Logistic Regression achieved stable convergence

7.2.1.3 Performed well on clearly worded user complaints

7.2.2 **Explainability**

7.2.2.1 Evidence retrieval successfully surfaces relevant conversational turns

7.2.2.2 Improves trust and interpretability of predictions

7.2.3 **System Responsiveness**

7.2.3.1 Lightweight inference enables near real-time response

7.3 Limitations

7.3.1 Keyword-based evidence extraction may miss semantically similar phrasing

7.3.2 No explicit test/train split reported for quantitative metrics

7.3.3 Session memory is limited to a single runtime instance

8. Database Design

8.1 Stored Fields

8.1.2 Query

8.1.2 Predicted Query Category

8.1.3 System Output (JSON)

8.1.4 Remarks

8.1.5 Timestamp (implicit via database schema)

8.2 Purpose

8.2.1 Enables auditing of predictions

8.2.2 Supports multi-turn conversational reasoning

8.2.3 Allows future offline evaluation

9. Conclusion

The implemented Causal Conversation Analysis System successfully integrates **machine learning-based intent prediction** with **evidence-backed explanations**.
Its modular architecture, explainable outputs, and persistent logging make it suitable for real-world conversational analytics tasks such as customer support analysis.

10. Future Enhancements

10.1 Replace keyword-based evidence extraction with semantic similarity models
10.2 Introduce train/test splits and quantitative evaluation metrics
10.3 Add transformer-based intent classification
10.4 Improve session handling for concurrent users
10.5 Extend dataset with more diverse conversational intents

TEAM HacKNomads:
GitHub Repository:
https://github.com/nayaksubham2426/HacKNomads_CausalAnalysisAndConversationalData

Deployment Link:
https://grahakmitra.vercel.app/