

Technical Report: Conversation Compliance & Quality Analyzer

1. Implementation Overview

We built a Streamlit app to analyze debt collection conversations stored in JSON format. Each utterance contains: **speaker**: Agent or Customer **text**: spoken content **stime** / **etime**: start and end timestamps

Question 1: Profanity Detection

- Pattern Matching: Regex patterns for profane words (e.g., fuck, shit, damn).
- LLM: Prompted GPT model with "Does this text contain profanity? Yes/No."

Question 2: Privacy & Compliance

- Pattern Matching: Regex detects if the agent discloses financial/account details (balance, due amount, account number). Also checks if verification (DOB, SSN, address) occurred first.
- LLM: Prompted GPT model with: "Does this agent statement disclose sensitive financial/account information before customer verification? Yes/No."

Question 3: Call Quality Metrics

- Computed deterministically:
- Silence = gaps between utterances.
- Overtalk = overlapping utterances.
- Visualized using bar charts (single call) and line plots (multi-call).

2. Comparative Analysis

- **Profanity**: Regex works best (faster, cheaper, more reliable).
- **Compliance**: Regex is limited, LLM handles nuanced cases better.
- **Metrics**: Numeric calculations — no ML needed.

3. Recommendations

- Use **Regex** for profanity detection (production-ready).
- Use **LLM** for compliance detection in real-world deployments.
- Continue using deterministic calculations for silence/overtalk.
- Hybrid approach is optimal: regex for clear patterns + LLM for subtle/ambiguous cases.

4. Visualization Findings

- Silence % highlights gaps in conversation flow (long pauses may indicate poor engagement).
- Overtalk % highlights interruptions (can indicate low professionalism).
- Visualizations help QA teams quickly compare call quality across agents.