Technical Report: Call Analysis System Evaluation

Executive Summary

This report evaluates the implementation of a call analysis system designed to detect profanity, privacy/compliance violations, and analyze call quality metrics in customer service conversations. The system employs dual approaches for detection tasks: pattern matching (regex-based) and Large Language Model (LLM) based analysis.

1. Implementation Recommendations

1.1 Profanity Detection (Question 1)

Current Implementation Analysis

The system implements two approaches for profanity detection:

1. **Pattern Matching (Regex)**:
    - Uses predefined lists of profanity patterns, obfuscated profanity, and contextual rudeness
    - Efficient with minimal computational overhead
    - No external API dependencies
2. **LLM-based Detection (Gemini API)**:
    - Leverages Google's Gemini API to analyze conversation context
    - Requires API key and internet connectivity
    - More resource-intensive and incurs API costs

Observations

Aspect	**Regex Approach**	**LLM Approach**
Accuracy	Good for explicit profanity; may miss novel obfuscations	Better contextual understanding; can detect subtle profanity
Efficiency	Very fast	Slower; dependent on API latency
Maintenance	Requires manual updates to profanity lists	Self-improving with model updates
Cost	Free; no API costs	Pay-per-use API costs
Adaptability	Limited to predefined patterns	Can adapt to new language patterns

Recommendations

1. **Hybrid Approach Implementation**:
    - Use regex as the first line of detection for obvious profanity
    - Only invoke the LLM for ambiguous cases where context matters
    - This would balance performance and cost
2. **Enhanced Regex Patterns**:
    - Expand the current regex patterns to include more regional variations
    - Add phonetic matching to catch more obfuscated profanity
    - Implement regular updates to the profanity lists from standardized sources
3. **Performance Optimization**:
    - Implement caching of LLM results for similar text patterns
    - Consider batch processing for offline analysis to reduce API costs

1.2 Privacy and Compliance Violation Detection (Question 2)

Current Implementation Analysis

Similar to profanity detection, the system uses two approaches:

1. **Pattern Matching (Regex)**:
    - Checks for sensitive information disclosure before identity verification
    - Uses extensive patterns for both sensitive information and verification requests
    - Works without external dependencies
2. **LLM-based Detection**:
    - Uses Gemini API to understand the contextual flow of the conversation
    - Can better assess the relationship between verification and disclosure

Comparative Analysis

Aspect	**Regex Approach**	**LLM Approach**
Accuracy	Good for explicit patterns; struggles with conversation flow	Better understanding of verification workflow and timing

False Positives	Higher rate	Lower rate; better contextual understanding
Explainability	Clear pattern matches can be highlighted	Decisions are harder to explain
Processing Speed	Very fast	API-dependent latency

Recommendations

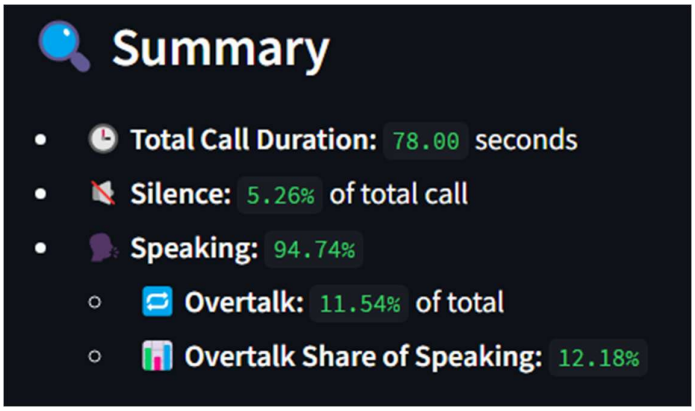
1. **Sequential Processing Model**:
    - Implement a stateful analysis that tracks conversation flow explicitly
    - Record verification events and subsequent sensitive information sharing
    - This would improve context understanding even with regex approach
2. **LLM Implementation Improvements**:
    - Add specific examples of compliance violations in the prompt
    - Request structured output that indicates which specific violation occurred
    - Implement confidence scores for detections
3. **Verification Tracking**:
    - Add a verification status tracker that maintains state across the conversation
    - Implement verification expiration logic for long conversations
    - Consider multi-factor verification detection

2. Visualization Analysis (Question 3)

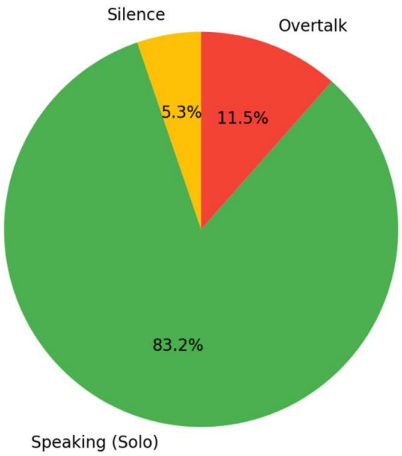
2.1 Current Visualization Implementation

The call quality metrics visualization includes:

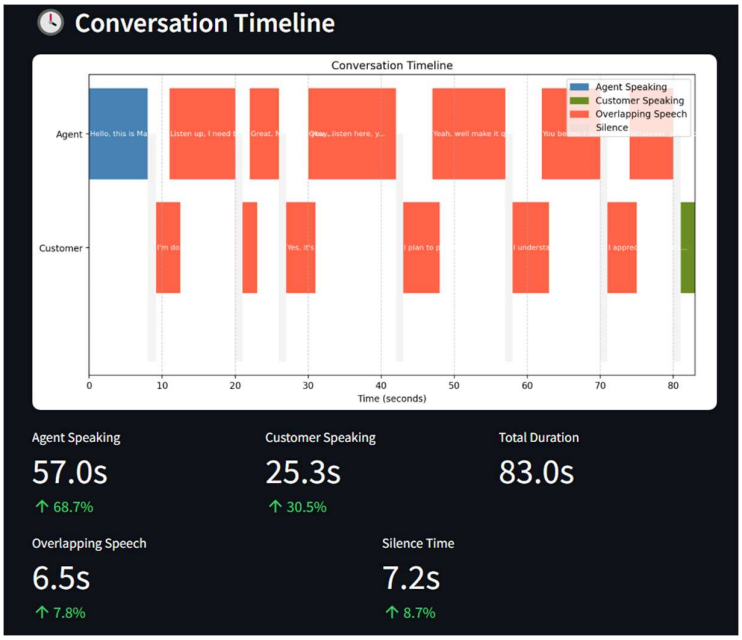
1. **Summary Statistics**:
    - Total call duration
    - Silence percentage
    - Speaking percentage
    - Overtalk percentage
    - Overtalk share of speaking time



2. **Pie Chart**:
    - Distribution of silence, solo speaking, and overtalk
    - Color-coded segments (yellow for silence, green for speaking, red for overtalk)



3. **Conversation Timeline**:
    - Horizontal timeline showing Agent and Customer speech segments
    - Color-coded bars (blue for agent, green for customer, red for overlapping speech)
    - Text annotations for longer speech segments
    - Silence visualization with gray bars



4. Conclusion

The current call analysis system provides a solid foundation for detecting profanity and compliance violations while visualizing call quality metrics. By implementing the recommended improvements, the system can achieve higher accuracy, better contextual understanding, and more insightful visualizations.

The hybrid approach combining pattern matching and LLM-based analysis offers the best balance between performance, cost, and accuracy. Enhanced visualization features will provide deeper insights into conversation patterns and help identify areas for improvement in customer service interactions.