Debt Collection Call Analysis: Technical Report

1. Introduction

This report presents a comparative analysis of approaches implemented for **Question 1 (Profanity Detection)** and **Question 2 (Privacy and Compliance Violation)** of the assignment. It aims to evaluate these methods based on results, performance, reliability, and usability, and to recommend the most effective approach. The document also includes a concise technical implementation guide to assist replication or adaptation of the solutions.

2. Comparative Analysis of Q1 and Q2 Approaches

Aspect	Question 1: Profanity Detection	Question 2: Privacy & Compliance Violation
Pattern Matching	Regex-based detection of profane words and phrases; fast execution, easy to implement, low computational cost.	Not explicitly implemented via regex due to complexity of contextual privacy violations.
LLM-based Approach	Uses large language models to detect subtle and contextual profanity, including mild and phrase-based cases.	LLM used for verifying complex compliance rules (verification before sensitive info sharing).
Hybrid Approach	Combines regex and LLM for pragmatic balance: fast detection + context sensitivity.	Pure LLM approach due to nuanced compliance logic.
Complexity	Regex simple, LLM integration requires API and prompt engineering.	High complexity—LLM well-suited for nuanced privacy checks.
Flexibility	Regex limited to known patterns; LLM generalizes better to novel inputs.	LLM best for policy and sequence-aware compliance identification.

Aspect	Question 1: Profanity Detection	Question 2: Privacy & Compliance Violation
Result Accuracy	Regex alone misses subtle profanity; hybrid enhances detection.	LLM provides reliable privacy violation detection with detailed explanations.
Performance	Regex faster; LLM slower and dependent on API latency and cost. Hybrid approach balances speed and accuracy.	LLM approach resource intensive but necessary for compliance context.
Usability	Hybrid approach integrated seamlessly in app, minimal user overhead.	LLM-based workflow requires environment setup (API key), but yields precise results.

3. Evaluation Criteria

- **Results**:

  Hybrid profanity detection effectively balances false negatives and false positives, outperforming regex alone. Privacy violation detection benefits from LLM's ability to understand complex context and sequence, producing clearer violation flags and explanations.

- **Performance**:

  Regex detection is very fast and lightweight. LLM invocations take longer and depend on external API availability and request limits. The hybrid model optimizes this by using regex as a first filter. Privacy checks inherently require LLM's nuanced understanding.

- **Reliability**:

  Regex-based methods are deterministic but limited by pattern coverage. LLM responses can vary but with well-tuned prompts provide robust context-aware outputs. Hybrid model enhances robustness. Privacy detection relies exclusively on LLM, ensuring compliance rules are interpreted appropriately.

- **Usability**:

  The implemented Streamlit app offers a clean UI enabling users to select appropriate entities and approaches. The hybrid detection requires minimal

configuration, while LLM-based privacy detection requires an API key set as an environment variable.

---

4. Recommendation

For Profanity Detection (Question 1):

The **hybrid regex + LLM approach** adopted in the app is recommended. It delivers a practical balance between speed, cost, and contextual accuracy by leveraging fast pattern matching and supplementing with intelligent LLM detection for subtle cases, maximizing detection quality.

For Privacy and Compliance Violation Detection (Question 2):

The **LLM prompt system alone is recommended** due to the complexity and subtlety of compliance requirements. Regex-based methods are not sufficient to capture the necessary context of verification order and privacy policy adherence. The LLM approach provides comprehensive reasoning and plain-language explanations suitable for compliance analysis.

---

5. Technical Implementation Guide

Question 1: Profanity Detection

1. **Load Conversation File:** Parse call transcript JSON to extract utterances.
2. **Regex Pattern Loading:** Load profanity word list, compile regex patterns.
3. **Profanity Detection per Utterance:**
   - Use regex matching to flag obvious profane words.
   - Optionally invoke LLM with prompt to detect contextual profanity in text.
   - Combine results from regex and LLM per utterance.
4. **Aggregate Results:** Summarize profane words detected per speaker and call.
5. **Output:** Provide a flag for presence or absence of profanity, detailed utterance-level outputs, and summary statistics.

Question 2: Privacy & Compliance Violation

1. **Load Conversation:** Parse JSON conversation file.
2. **Configure LLM Client:** Supply API key and load compliance prompt.
3. **LLM Invocation:** Pass formatted conversation text to LLM prompt designed to detect privacy violations based on verification sequence and sensitive info sharing.
4. **Parse LLM Response:** Extract violation flag, explanation, and related terms.
5. **Output:** Return compliance violation indicator, explanatory summary, and related keywords.

---

6. Conclusion

This report highlights that the hybrid regex and LLM approach is optimal for profanity detection balancing accuracy and efficiency. For privacy compliance detection, the nuanced nature of the task necessitates a dedicated LLM prompt system. Together, these approaches meet the assignment’s goals with a clear separation of methods by task complexity, supported by a user-friendly Streamlit application. This dual-strategy ensures reliable, interpretable call analysis suitable for regulatory and operational use.
