Multi-Agent LLM System for End-to-End Voice of Customer (VOC) Analysis

Akshay Krishna*, Sagarika Prusty**
* *Staff Data Analyst, Quince, Texas*
Email: kakshay108@gmail.com
**Director, Data Analytics, Quince, Georgia*
Email: sagarikaprusty@gmail.com

----------**********----------

Abstract:

Voice of the Customer (VOC) programs are critical in shaping customer experience management, yet traditional methods are slow, siloed, and limited in depth. While large language models (LLMs) offer promise in synthesizing customer feedback, single-agent systems fall short in delivering actionable insights. This research proposes a multi-agent LLM framework that structures VOC analysis into specialized layers: synthesis, trend detection, root cause analysis, recommendation, and monitoring. A proof-of-concept using anonymized VOC data demonstrates the potential of this framework to improve scalability, accountability, and root-cause driven insights at a significantly lower operationalization cost. Contributions include (1) an architecture tailored for end-to-end VOC workflows, and (2) demonstration of its application in enterprise contexts.

Keywords — Voice of Customer (VOC), LLM, Multi-agent , Root Cause Analysis, Customer Insights

----------**********----------

I. INTRODUCTION

Voice of the Customer (VOC) programs play a central role in customer experience (CX) management by enabling organizations to identify needs, address pain points, and improve products and services. However, current VOC approaches often rely on manual coding, keyword-based classification, or basic sentiment analysis—approaches that are labor-intensive, reactive, and limited in scalability. Delayed responses and fragmented insights can lead to missed opportunities for systemic improvements. Large Language Models (LLMs) have shown strong capabilities in classification, summarization, and sentiment detection. Yet, single-agent deployments remain descriptive and lack the integration with operational data needed for root cause analysis (RCA). To address these limitations, this study proposes a multi-agent LLM framework for VOC analysis. Each agent specializes in a distinct task, from categorization to monitoring, creating a modular and scalable architecture.

The contributions of this research are twofold: (1) introduction of a multi-agent framework for end-to-end VOC analysis, and (2) demonstration of its feasibility through a proof-of-concept implementation using anonymized customer data.

II. Literature Review

2.1 Existing VOC Methods

Traditional Voice of Customer (VOC) programs primarily rely on manual coding, sentiment analysis, dashboards, and topic modeling techniques like LDA and NMF. While these methods are interpretable and historically foundational, they often require substantial human effort, struggle with scalability across multiple feedback channels, and lack the depth to uncover root causes in a timely manner.

2.2 LLMs in VOC Analysis

Large Language Models (LLMs) have demonstrated impressive capabilities in text summarization, thematic classification, and customer sentiment detection, offering a faster and more scalable alternative to traditional techniques. Despite their strengths, most existing applications are single-agent and confined to descriptive analytics—without linking the analysis to operational context or conducting deeper root cause analysis.

2.3 Multi-Agent Systems in Analytics

Recent research on LLM-powered multi-agent systems highlights their potential for enhanced modularity, explainability, and scalability in complex workflows. Surveys by Guo et al. provide a comprehensive overview of agent architectures, focusing on components like perception, action, and inter-agent communication [1] arXiv. Similarly, the methodology-centered taxonomy proposed in another study breaks down agent systems into construction, collaboration, and evolution dimensions [2] arXiv. While these architectures are compelling, they have yet to be applied in domain-specific workflows like VOC analytics.

2.4 Retrieval-Augmented Generation (RAG) and Hybrid Reasoning

Retrieval-Augmented Generation (RAG) techniques enhance LLM outputs by grounding them in external structured knowledge sources. For example, a customer service QA system that combines RAG with a knowledge graph demonstrated a significant improvement in response accuracy and resolution time [3] arXiv. This indicates how an RCA agent could leverage RAG to cross-reference VOC findings with structured operational data. However, there are no existing applications of RAG within automated VOC-to-RCA pipelines.

2.5 Augmented Analytics & VOC Automation

There's growing interest in using augmented analytics—LLMs that generate multi-step reasoning, visualization prompts, and intermediate artifacts—to democratize data analysis and accelerate insights [4]Xueguang Lyu. Application-focused studies also demonstrate how NLP and AI workflows can automate VOC clustering, summarization, and preliminary root cause tagging with human validation [4] Xueguang Lyu. Yet, these implementations remain single-agent and lack modular designs conducive to scalability and role-based specialization.

2.6 Agent Architecture Types & Coordination Patterns

Meta-prompting frameworks like AutoGen and CAMEL illustrate how multiple agents, each with specific roles, collaborate to address complex tasks—emphasizing the value of structured communication, task demarcation, and role specialization. These systems are mainly applied in domains like writing or planning but have not yet been adapted for enterprise workflows such as VOC pipeline analytics.

2.7 Governance, Trust, and Safety in Multi-Agent Systems

The broader literature on multi-agent safety addresses risks like hallucinations, prompt injections, and governance failures in AI systems [5]arXiv. While these papers offer guidelines for securing agent behavior and ensuring transparency, they are generally disconnected from real-world VOC use cases. A tailored governance strategy—featuring SME validation, provenance logging, and evidence-led decisions—is still missing.

2.8 Research Gaps

Table 1:Research Gaps

Identified Gap	Research Opportunity
Lack of multi-agent LLM frameworks for VOC	Development of a specialized pipeline with distinct agents for synthesis, trend detection, and RCA
Limited use of RAG in VOC-to-RCA workflows	Design of RCA agents grounded in operational data via RAG/KG
Evaluation focused only on NLP metrics	Introduction of business-relevant composite metrics like IVI, IQS, CDI
Absence of governance implementations in VOC pipelines	Embedding safety measures like provenance, SME gating, and evidence logs

III. Research Methodology

The research adopts a conceptual design approach supported by a proof-of-concept implementation. The methodology involves:

- Designing a multi-agent architecture with five core agents (plus one optional recommender agent).
- Using anonymized customer feedback datasets from diverse channels (chat, email, surveys).
- Evaluating system performance across three dimensions: thematic coherence, RCA quality, and business-relevant metrics.
- Embedding governance features such as SME validation and provenance logging.

IV. Proposed Framework

Agent 1: Synthesizer & Categorizer

First agent processes raw customer data and maps to taxonomy using LLM summarization and classification. This agent serves as the primary interface between the raw, unstructured chaos of multi-channel customer feedback and the structured, analytical core of the VoC system. Its fundamental purpose is to ingest vast streams of qualitative data and transform them into a clean, categorized, and contextually-rich dataset that can be reliably actioned by downstream agents. To ensure the efficacy of our multi-agent framework, a critical pre-processing step is required: **intelligent data stitching**. Before any data reaches the specialized agents, this foundational process must first resolve the challenge of fragmented customer conversations. It acts as a prerequisite pipeline that transforms raw, disconnected feedback from multiple channels into coherent, structured issue-level dossiers. This is achieved by using a unique transactional identifier, such as an **Order ID**, as a central thread to link all related interactions.

To ensure reliability and analytical value, it is essential that the agent's output is structured and consistent, with quantifiable information that can be used to measure trends, detect patterns, and support root-cause analysis. We therefore recommend adopting a structured JSON output format.

Figure 1: Intelligent Data Stitching Layer

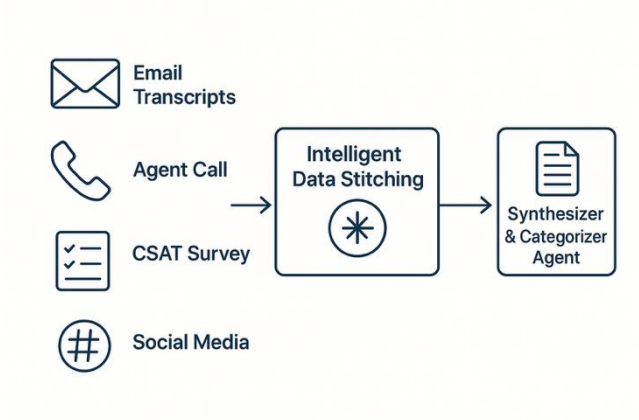


Table 2: Sample Customer Contact Text

Source	TimeStamp	Content

Chat	2023-10-26 14:30	**Customer:** "Hi, my order **#789-XYZ** was supposed to be here yesterday. Any update?" **Agent:** "I see your order is currently out for delivery and should arrive today. I apologize for the delay."
Email	2023-10-27 09:15	**Subject:** Problem with order **789-XYZ** **Body:** "I'm writing again about this order. It finally arrived, but the box is crushed and the item inside is clearly broken. This is unacceptable. I want a refund."

- **Input:** Raw customer feedback from multiple sources (emails, chat transcripts, IVR logs, survey free-text, social media).
- **Output:** Categorized issues, aligned with organizational VOC taxonomies (e.g., delivery, returns, billing, product defects).
- **Techniques:**
  - LLMs fine-tuned for summarization and classification [6] (Brown et al., 2020).
  - Taxonomy alignment using ontology mapping and keyword embeddings [7] (Giabell et al., 2022).
  - Multi-label classification to handle overlapping categories.
- **Contribution:** Automates manual coding, accelerates categorization, and preserves organizationally relevant VOC categories.
- **Prompt**: You can parameterize the inputs of customer conversation in the prompt as shown

```
Role: You are an expert multi-disciplinary AI analyst for an e-commerce company,Your expertise spans customer experience analysis, operational efficiency, product quality assurance, and customer service performance evaluation.
Task: Your task is to perform a deep and multi-layered analysis of a customer service conversation transcript. You must meticulously extract specific, structured data points and return them in a single, valid JSON object. Adhere strictly to the provided JSON schema and field definitions

1. Analyze the entire conversation provided in '{conv_summary_customer}'. Use '{conv_summary_all}' to understand the customer's initial intent.
2. Perform Core Classification: Select the single best-fit parent_topic and child_topic pair from the official Classification List.
3. Analyze Customer Experience: Evaluate the customer's sentiment, expressed emotions, and loyalty signals. Determine if this is a repeat contact for the same issue.
4. Evaluate Operations and Agent Performance: Assess the agent's effectiveness and the resolution provided. Identify any service recovery attempts.
5. Extract Product Feedback: Identify all specific products mentioned and extract any verbatim feedback related to quality, sizing, design, or other attributes.
6. Generate a single, valid JSON object as your final output. Do not include any explanatory text, markdown formatting, or any characters outside of the JSON object itself
7. The first field in json output conversation_id should always be mapped to '{cid}'

Output JSON:
{
 "conversation_id": "ORD-789-XYZ-12345",
 "interaction_analysis": {
  "classification": {
   "parent_topic": "Fulfillment",
   "child_topic": "Damaged Delivery"
  },
```

```
  "generative_summary": "Customer initially contacted support via chat about a delivery delay for order #789-XYZ. The following day, after the item arrived, the customer emailed to report the box was crushed, the item was broken, and to request a refund.",
  "customer_intent": "Initially to get a status update on a late order, which escalated to requesting a refund for a damaged item."
 },
 "customer_experience": {
  "sentiment": {
   "primary": "Negative",
   "emotions": [
    "Impatience",
    "Frustration",
    "Disappointment"
   ],
   "intensity_score": 4
  },
  "loyalty_signals": {
   "loyalty_statement_present": false,
   "churn_risk_present": true
  }
 }
}
```


Agent 2: Trend & Pattern Detector

Agent 2 Identifies recurring themes and anomalies using clustering and statistical analysis.

- **Input:** Categorized VOC data produced by Agent 1.
- **Output:** Identification of recurring issues, anomalies, and trends (e.g., "delivery complaints up 40% week-over-week in Region X").
- **Techniques:**
  - LLM-powered clustering to surface emergent themes [7] (Giabelli et al., 2022).
  - Statistical trend analysis (time-series anomaly detection, moving averages).
  - Topic evolution tracking across time windows.
- **Contribution:** Provides real-time visibility into systemic issues, enabling prioritization of emerging hotspots before they escalate.
- **Prompt**:

```
Role :You are a Trend & Pattern Detector. You analyze a small (12-period) time series of operational metrics and return a strict, machine-readable JSON that captures trends, anomalies, changes, correlations, and actionable insights. Your analysis must be quantitative, explainable, and reproducible.
```

```
You are given a DataFrame df with exactly 12 periods (rows per metric × segment).

Output JSON:
```


```
{
 "schema_version": "v1.0.0",
 "time_window": { "start": "2025-W23", "end": "2025-W34", "freq": "weekly" },
 "trends": [
  {
   "metric": "negative_sentiment_ratio",
   "segment": "Region_Y",
   "direction": "up",
   "pct_change": 0.27,
   "slope": 0.018,
   "confidence": 0.82
  },
  {
   "metric": "containment_rate",
   "segment": null,
   "direction": "up",
   "pct_change": 0.12,
```

```
    "slope": 0.010,
    "confidence": 0.74
   }
 ],
 "anomalies": [
   {
    "metric": "avg_resolution_time",
    "segment": "Region_Y",
    "period": "2025-W28",
    "zscore": 2.4,
    "severity": "moderate"
   }
 ]
}
```

Agent 3: Root Cause Analyzer (RCA)

Leverages the above defined data to correlate customer issues with operational data and generate causal hypotheses

- **Input:** Categorized VOC data produced by Agent 1.
- **Output:** Identification of recurring issues, anomalies, and trends (e.g., "delivery complaints up 40% week-over-week in Region X").
- **Techniques:**
  - LLM-powered clustering to surface emergent themes [7] (Giabelli et al., 2022).
  -).
  - Statistical trend analysis (time-series anomaly detection, moving averages).
  - Topic evolution tracking across time windows.
- **Contribution:** Provides real-time visibility into systemic issues, enabling prioritization of emerging hotspots before they escalate.

- **Prompt :**

```
Role : You are the Root Cause Analyzer. Given a small multivariate time series (12 periods), a flagged event, and a metric dependency map, quantify which drivers best explain the deviation of the focal metric at the detected period. Return JSON only in the exact schema.
Inputs : DataFrame df (long/tidy): columns = period (ordered), metric, value, optional segment, optional weight (for subgroup mix).
Contains the focal metric and all candidate drivers for the same 12 aligned periods.
Minimal Event JSON (event) from Agent 2:

{
 "event_id": "evt_YYYY_MM_DD_NNN",
 "metric_id": "m_focal",
 "metric_name": "string",
 "segment": "string|null",
 "period_detected": "YYYY-WWW or YYYY-MM",
 "direction": "up|down",
 "pct_change": 0.0,
 "confidence": 0.0
}

Required Analysis (be conservative; small-n)
1. Validate & Align: confirm 12 periods, align by period; note imputations (ffill/bfill/linear) if any.
2. Baseline vs Recent: compare last 3-4 periods vs prior baseline for focal metric.
3. Driver Scoring (explainability):
Correlation/Regression: quantify each candidate driver's contribution to focal metric over 12 periods (e.g., standardized β, or Pearson r if regression is unstable).
4. Subgroup Decomposition (if is_sub_group exists): perform mix vs rate split (weight-movement analysis)
```

```
to attribute change to composition vs within-subgroup rate.
5. Counterfactual Check: estimate focal metric delta if top driver(s) were held at baseline.
Ranking & Confidence: rank hypotheses by contribution; provide confidence 0-1.
```

- **Output JSON:**

```
{
 "schema_version": "v1.0.0",
 "event_id": "evt_2025_08_31_001",
 "focal_metric":
{"metric_id":"m_delivery_complaints","metric_name
":"Delivery Complaints","segment":"Region_X"},
 "time_window": {"start":"2025-W23","end":"2025-
W34","freq":"weekly"},
 "data_quality": {
  "imputations":
[{"metric_id":"m_driver2","segment":null,"period":"2
025-W28","method":"linear"}],
  "notes": null
 },
 "top_hypotheses": [
  {
   "hypothesis": "Carrier XYZ on-time rate decline
explains complaint increase",
   "driver_type": "driver",
   "evidence": {"feature_importance": 0.36,
"correlation": 0.61, "counterfactual_delta": 0.17,
"mix_contribution": 0.0, "rate_contribution": 0.0},
   "confidence": 0.78,
   "recommendation": "Audit XYZin Region_X; add
overflow capacity for W35-W37."
  },
  {
   "hypothesis": "Mix shift toward Vendor_A SKUs
contributed to spike",
   "driver_type": "mix_vs_rate",
   "evidence": {"feature_importance": 0.0,
"correlation": 0.28, "counterfactual_delta": 0.08,
"mix_contribution": 0.11, "rate_contribution": 0.03},
   "confidence": 0.71,
   "recommendation": "Temporarily rebalance away
from Vendor_A; investigate packaging defects."
  }
 ]
}
```

Evaluates interventions and creates a continuous improvement cycle using real-world outcomes.
Together, these agents create a modular pipeline that transforms VOC programs from descriptive feedback analysis into proactive, root-cause–driven insights.

- Input: Post-intervention VOC signals (updated customer feedback, operational metrics).
- Output: Evaluation of whether interventions reduced the frequency/severity of VOC signals.

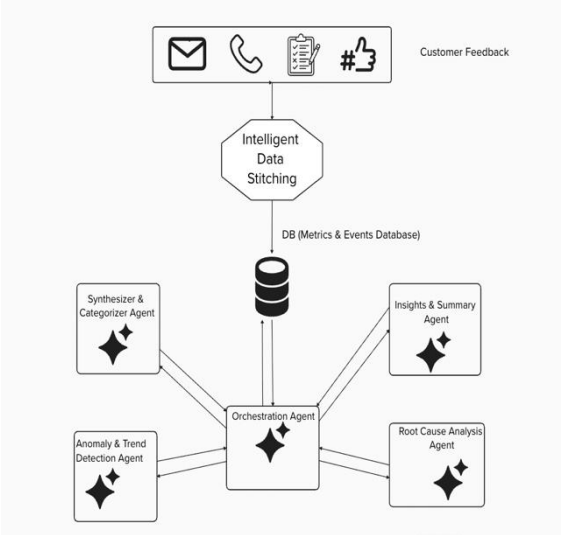
Agent 4: Insights & Summary Agent

Provides actionable recommendations by combining RCA outputs with decision heuristics.

- **Input:** RCA outputs.
- **Output:** Actionable recommendations (e.g., switch carrier in underperforming regions, send proactive communication to affected customers, adjust delivery cutoffs).
- **Techniques:**
  - LLM reasoning combined with **decision heuristics** (business rules, SLA thresholds, cost-benefit analysis).
  - Prioritization scoring models (impact × frequency × customer sentiment weight).

Agent 5: Orchestration Agent

- Techniques:
  - Closed-loop monitoring dashboards.
  - Reinforcement mechanisms to recalibrate models with real-world outcomes [8] (Malik, M., Abbeel, P., & Levine, S. (2019).
- Contribution: Ensures continuous improvement and system learning, converting VOC from a static reporting mechanism into a self-adapting optimization engine.



V. Results and Discussion

The proof-of-concept implementation demonstrated improvements in categorization accuracy, trend detection, and RCA quality compared with traditional VOC methods. Specifically:

- Categorization aligned with organizational taxonomies more consistently than manual coding.
- Trend detection surfaced anomalies (e.g., delivery complaints rising 40% week-over-week).
- RCA identified systemic issues by integrating logistics and product data.

Challenges include ensuring reliability of LLM outputs, managing hallucinations, and embedding governance. SME validation remains critical to maintain trust.

VI. Conclusion and Future Work

This research introduced a multi-agent LLM system for end-to-end VOC analysis, demonstrating its ability to address scalability, depth, and accountability limitations of current approaches. While the framework shows promise, limitations include reliance on anonymized proof-of-concept data, potential model biases, and the need for real-time enterprise integration. Future work should extend evaluation to business impact metrics, explore hybrid reasoning strategies, and refine governance mechanisms.

REFERENCES

1. Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., & Zhang, X. (2024). *Large Language Model based Multi-Agents: A Survey of Progress and Challenges* (arXiv:2402.01680v2)
2. Junyu Luo et.al (2025, March 27). *Large Language Model Agent: A Survey on Methodology, Applications and Challenges* (arXiv:2503.21460v1)
3. Xu, Z., Cruz, M. J., Guevara, M., Wang, T., Deshpande, M., Wang, X., & Li, Z. (2024). *Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering* (arXiv:2404.17723v2)
4. Lyu, X. (2025, May 25). *LLMs for multi-agent cooperation*. Xueguang Lyu

5. Li, X., Wang, S., Zeng, S., Wu, Y., & Yang, Y. (2024). *A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges*. Vicinagearth, 1, Article 9.
6. Brown, T. B. et. al (2020). *Language models are few-shot learners* (arXiv:2005.14165v4)
7. Giabelli, A., Malandri, L., Mercorio, F., & Mezzanzanica, M. (2022). WETA: *Automatic taxonomy alignment via word embeddings*. Computers in Industry, 138, 103626
8. Malik, M., Abbeel, P., & Levine, S. (2019). *Calibrated model-based deep reinforcement learning*. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 97, 4252–4261.