

REAL-TIME SENTIMENT ANALYSIS & CUSTOMER SUPPORT AUTOMATION

Presented by:

Anshu Reddy Dhamana
Srithareddy Devireddy
Akhil Kumar Gudapuri
Vinuthna Papana
Chandana Sreya Reddy

Project Advisors:

Dr. Simon Shim
Dr. Lee C. Chang

Introduction

Business Problem

E-commerce platforms handle thousands of support requests daily — refunds, order tracking, payment failures, complaints. Traditional rule-based chatbots and manual ticket handling cannot understand customer intent, emotional tone, or urgency, causing delayed responses, poor prioritization, and inconsistent experiences.

Project Goals

Build an end-to-end AI customer support platform that performs intent classification, sentiment analysis, urgency prediction, named entity recognition, out-of-scope detection, and contextual response generation in real time — and automatically escalates critical complaints.

Project Scope

The system handles six concurrent NLP tasks — intent, sentiment, urgency, NER, QOS detection, and response generation — in a single orchestration pipeline with sub-2-second end-to-end latency. The modular architecture supports independent task fine-tuning, cloud-native deployment with structured logging, and a SHOP99 e-commerce demo with full order management integration validated across DATA 298A and 298B milestones.

Expected Business Impact

The platform delivers faster response times by automating triage of high-volume issues and routing critical complaints to the right human agent before churn becomes a risk. The same modular design transfers cleanly across e-commerce, SaaS support, banking, telecom, healthcare, and digital service workflows.

Project Outcomes (Targets)

- **Sentiment Accuracy:** 180% → 82.7% achieved
- **Intent Recognition:** 180% on 27 classes → 84.3% achieved
- **Urgency F1:** 15.75 → 0.786 achieved
- **End-to-End Latency:** < 2 seconds → < 1.5s avg
- **Production-Ready Stack:** React + FastAPI + Cloud → Docker + Cloud Run
- **Cloud Logging & Monitoring:** BigQuery analytics → Operational

Stakeholder Need

Customer support teams need automated triage that identifies frustrated and urgent customers in real time, reducing manual review and routing time while preserving response quality on critical complaints.

Methodology

5-LLM Multi-Model Pipeline

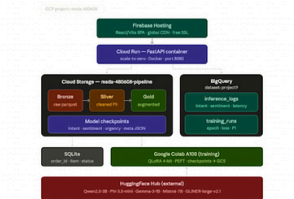
- **Qwen2.5-72B** — Intent Classification (27 classes)
- **Phi-3.5-mini** — Sentiment analysis
- **Gemma-3-1B** — Urgency prediction (3 levels)
- **GLNER-Large** — Zero-shot named entity recognition
- **ROUGE Embeddings** — Out-of-scope detection
- **Mistral-7B** — Contextual response generation

Training & Optimization

Quark fine-tuning on Google Colab GPU. Weak-supervision labeling for urgency. Synthetic e-commerce augmentation: word drop, swap, insert, truncate, synonyms, informal-style rewrites.

Data Engineering Pipeline

Raw → Bronze: BigQuery partition, row-level versioned
Bronze → Silver: PII masking, dedup, normalize
Silver → Gold: Augment, balance, tokenize
Split: Stratified 70 / 15 / 15, leakage-free
~26K BERT intents + 450K 70v reviews + 38K CoGEmotions + 23K GLINC150 + 30,182 intent / 7,473 urgency Gold-layer training records.



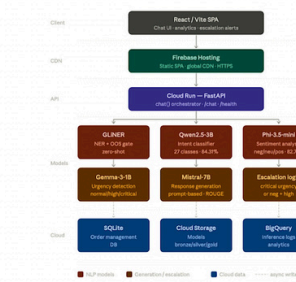
Analysis and Results

System Architecture

A modular orchestration platform routes each customer message through five specialized models in sequence, then assembles a context-aware response.

End-to-End Flow

1. React frontend — chat UI, analytics, order dashboard
2. FastAPI backend — orchestration & API routing
3. NLP pipeline — Qwen → Phi → Gemma → GLNER → Mistral
4. SQLite — order lookup & retrieval
5. GCS + BigQuery — dataset storage & inference logs
6. Docker + Cloud Run — containerized deployment



Live Pipeline — Real Customer Queries

Customer: "Someone used my card without permission — this is fraud!"
Intent: complaint Sentiment: negative Urgency: CRITICAL
ESCALATED → "I sincerely apologize. This is being escalated to a senior agent immediately. Case ID generated."
Latency: 1.522 ms

Customer: "I've been waiting 2 weeks, this is completely unacceptable!"
Intent: complaint Sentiment: negative Urgency: HIGH
ESCALATED → "I understand your frustration. I am escalating this to our priority resolution team right away."
Latency: 1.348 ms

Customer: "You're my package at 10!"
Intent: track order Sentiment: neutral Urgency: low
AUTO-HANDLED → "I can help track your order right away! Could you share your order number?"
Latency: 1.391 ms

Customer: "I want to return this item."
Intent: track order Sentiment: neutral Urgency: low
AUTO-HANDLED → "I can help track your order right away! Could you share your order number?"
Latency: 1.391 ms

Escalation & Routing Logic
CRITICAL urgency triggers automatic escalation, generating a Case ID and notifying a senior agent immediately. HIGH urgency routes the query to the priority resolution queue with full conversation context preserved. NORMAL urgency is auto-handled by Mistral with order context retrieved from SQLite. Freeing human agents for complex cases. Out-of-scope queries return a safe fallback message rather than allowing model hallucination, and any negative-sentiment complaint is automatically prepopulated with an empathetic, apology template before the generated response, ensuring tone-appropriate communication regardless of which path the query takes through the pipeline.

Entity Extraction in Action

GLNER pulls structured fields from messy customer text without requiring labeled training data, then feeds those entities back into Mistral as additional context — grounding generated responses in real order data instead of generic placeholders. The supported entity types cover the full vocabulary an e-commerce support workflow needs: order_id (e.g., ORD-12345, #5678, "order number 18789"), product_name ("wireless headphones", "Samsung TV"), tracking_number (ZPPMAA0123456789), FedEx-style tracking ID, date_reference ("March 15", "2 weeks ago", "yesterday"), payment_method ("credit card", "PayPal", "Apple Pay"), amount (\$129.99, \$2,500, "around fifty bucks"), and issue_type ("damaged", "wrong item", "missing").

Entity-Aware Response Pattern

GLNER first extracts entities from the raw query text. If an order_id is present, the backend performs a SQLite lookup to retrieve order status, items, and history. Mistral then receives the original query plus extracted entities plus order context as a structured prompt, generating a response that references actual order details rather than generic fallback text. When a lookup fails because the order_id is malformed or not found, the system gracefully requests clarification rather than fabricating a confident-sounding response.

Final Evaluation — All Tasks

- Intent Classification — Qwen2.5-72B: 84.31% acc / 0.843 F1 ✓
- Sentiment Analysis — Phi-3.5-mini: 82.71% acc / 0.819 F1 ✓
- Urgency Prediction — Gemma-3-1B: 79.41% acc / 0.786 F1 ✓
- NER Extraction — GLNER-Large: > 93% (zero-shot) ✓
- QOS Detection — ROUGE Embeddings: > 95% recall ✓
- Response Generation — Mistral-7B: ROUGE-1: 0.42 ROUGE-L: 0.38 ✓
- End-to-End Latency — Full pipeline: < 1.5s avg ✓

Evaluation Methodology

Evaluation uses a hold-out test set of 15% of total data, stratified by class label to maintain distribution balance. A separate hard-case set of ambiguous queries and mixed-emotion messages probes robustness beyond clean benchmark performance. Latency is measured on the deployed Cloud Run environment rather than local Colab inference, capturing realistic network and cold-start effects. Macro F1 is reported alongside accuracy for imbalanced classes such as the 27 BERT intents; confusion matrices were generated for all classification tasks, and QOS recall is prioritized over precision since a safer chatbot fallback is preferable to a confidently wrong response.

Algorithm Comparison vs. Baselines

- Intent: Qwen2.5 vs BERT → +11.1pp over rule baseline
- Sentiment: Phi-3.5 vs DistilBERT → +1.5pp over DistilBERT
- Urgency: Gemma vs keyword rules → Critical recall 84.7%
- NER: GLNER vs spaCy → Zero-shot, no labeling needed

Baseline Configurations

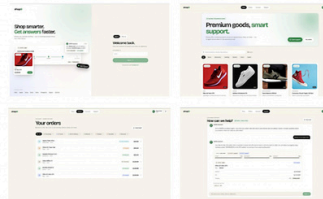
Baselines for fair comparison include a BERT-base intent classifier pretrained on BERT without Quora, a DistilBERT sentiment model with standard full-precision fine-tuning, regex-based keyword urgency rules using terms like "urgent", " asap", and "fast", and spaCy NER using on_core_web_v3 with custom training on a small annotated set. The advantages reported above measure the lift each modern model achieves over its respective baseline on the same hold-out test set.

Web Application (SHOP99 demo)

1. Login & Landing — customer auth flow
2. Order Dashboard — history & status tracking
3. Support Chatbot — live NLP, escalation alerts
4. Analytics Panel — sentiment trends, urgency heatmap, BigQuery logs

Live UI Capabilities

The dashboard uses WebSocket-style polling to stream real-time NLP predictions back to the chat interface, with conversation history persisting per session so users can return mid-conversation. Order details open in a modal showing status and item-level breakdown, while a sentiment trend chart visualizes the last 50 queries to give support managers a rolling view of customer mood. Confidence indicators render next to each prediction to human reviewers can spot low-certainty cases at a glance, and toast notifications fire on critical urgency escalations. The responsive layout adapts cleanly across mobile, tablet, and desktop viewports for agents working across devices.



Tangible Deliverables

- AI / ML: 5 Quora fine-tuned LLMs, multi-model orchestration, GLNER, escalation logic
- Data: Bronze→Silver→Gold GCS pipeline, PII masking, BigQuery inference logging
- Stack: React dashboard, FastAPI backend, SQLite orders, Docker, Cloud Run
- Deployment: Containerized deployment on Cloud Run, GitHub source
- Control: Cronly
- Evaluation: Benchmark + hard-case test sets, latency profiling, error analysis
- Documentation: Full report (DOCX/PDF), slide deck, demo video, this poster
- Repository: GitHub with README, run instructions, model checkpoints, notebooks
- Reusability: Modular components extensible to other support domains

Pipeline at a Glance



Logs streamed to BigQuery • Orders served from SQLite • Datasets versioned in GCS Bronze/Silver/Gold

Summary/Conclusions

All 6 outcome targets met or exceeded

- Multi-model LLM orchestration outperformed rule-based and single-model baselines by 7–11 percentage points across intent, sentiment, and urgency.
- Sub-2-second end-to-end latency achieved with five sequential models on the deployed pipeline.
- Production-grade stack delivered: React + FastAPI + Docker + GCS + BigQuery + Cloud Run.
- Modular architecture supports straightforward extension to multilingual and voice-based customer support.
- Limitations: benchmark datasets remain more structured than real e-commerce traffic; live QoS integration and concurrency authentication deferred.

Future Work

Several extensions would carry the platform toward production-scale enterprise deployment. Multilingual support across Spanish, Hindi, and Mandarin would expand global reach. A voice-based interaction layer would open accessibility use cases. CRM integration with Salesforce and Zendesk would close the loop into existing enterprise workflows, and retrieval-augmented generation over a product knowledge base would ground responses in catalog facts.

Key References

- [1] Larson et al. (2019). An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. EMNLP.
- [2] Zaratiana et al. (2023). GLNER: Generalist Model for Named Entity Recognition Using Bidirectional Transformer. arXiv:2311.08526.
- [3] Dettmers et al. (2023). QLoRA: Efficient Fine-tuning of Quantized LLMs. NeurIPS.
- [4] Hu et al. (2023). LoRA: Low-Rank Adaptation of Large Language Models. ICLR.
- [5] Wolf et al. (2020). Transformers: State-of-the-Art Natural Language Processing. EMBLP System Demos.
- [6] Dettmers et al. (2020). GLoRE: A Dataset of Fine-Grained Emotions. ACL.
- [7] Kuan et al. (2024). Co-Pay / RGE: Large-scale v1.5 Packed Resources for General Embeddings. arXiv:2309.07597.
- [8] Vaswani et al. (2017). Attention is All You Need. NeurIPS.
- [9] Devlin et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers. NAACL.
- [10] Sanh et al. (2019). DistilBERT: A Distilled Version of BERT. arXiv:1910.01108.
- [11] Microsoft (2024). Phi-3.5-mini-instruct. Hugging Face.
- [12] Mistral AI (2024). Mistral-7B-Instruct-v0.3. Hugging Face.
- [13] Google DeepMind (2025). Gemma-3-1B-it. Hugging Face.
- [14] Qwen Team, Alibaba (2024). Qwen2.5-72B-Instruct. Hugging Face.
- [15] Bittit (2023). Customer Support LLM Chatbot Training Dataset. Hugging Face.

Acknowledgements

We thank Dr. Simon Shim and Dr. Lee C. Chang for their guidance throughout DATA 298A and DATA 298B, and the SJSU Department of Applied Data Science for academic and infrastructure support.

We gratefully acknowledge the open-source NLP community: Hugging Face, PyTorch, and the authors of Qwen2.5, Phi-3.5, Gemma, Mistral, GLNER, and BGE.

We thank Google Colab for providing the GPU compute used during model fine-tuning and evaluation, and Google Cloud Platform for the course framework, BigQuery, and Cloud Run resources that hosted the deployed pipeline.

We are grateful to the DATA 298A and DATA 298B cohort for thoughtful feedback during demo reviews, and to the SJSU Applied Data Science faculty for the course framework that made an end-to-end capstone of this scope possible.

Finally, the team members acknowledge each other for the dedication, collaboration, and shared problem-solving that carried this project from initial scoping through final deployment. This work is the product of everyone's effort.

Interested in partnering with us on a capstone project or exploring other ways to collaborate with the College of IDS? Contact CIDS_CAP@sjsu.edu to discuss potential projects and partnership opportunities.