RESEARCH ARTICLE

JENER Journal of Empirical and Non-Empirical Research
Volume 2, Issue 5, Page 524-528 | Article No. 63
DOI: 10.4898/jener.v2i5.a63
ISSN: 7895-9216
Journal homepage: https://jenerjournal.com
Journal email: editor@jenerjournal.com

CLARA: A ChromaDB-Leveraged Automated Response Architecture for IT Helpdesk Support Using Lightweight Sentence Transformers and Open-Weight Language Models

J Pradeep Krishna[1], Pooja A[2] & Dr. V Rajalakshmi[3]

[1,2,3]Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Anna University, Chennai – 600 025, India

ARTICLE INFORMATION

Article history:
Published: May 2026

Keywords:
ChromaDB
IT Helpdesk Automation
Retrieval-Augmented Generation
ChromaDB
Semantic Vector Retrieval
RAGAS Evaluation

ABSTRACT

Sustained productivity in enterprise IT helpdesk operations depends on an agent's capacity to rapidly surface accurate resolution procedures from large, heterogeneous knowledge repositories. Conventional lexical-matching retrieval integrated into commercial ticketing platforms consistently underperforms when ticket descriptions employ informal phrasing, domain-specific abbreviations, or synonym-rich language that diverges from indexed documentation vocabulary. This paper proposes CLARA (ChromaDB-Leveraged Automated Response Architecture), a modular and fully open-weight pipeline that addresses this retrieval shortfall by combining semantic vector search with generative language modelling to produce ready-to-dispatch resolution draft responses for helpdesk agents. CLARA ingests support tickets directly from a designated Gmail mailbox via the Gmail API, encodes incoming ticket text and indexed knowledge documents using the all-MiniLM-L6-v2 sentence transformer, maintains dual-partition persistent vector storage in ChromaDB, and synthesises contextually grounded response drafts using Llama 3.1 8B under 4-bit quantisation. The pipeline is evaluated on 3,200 enterprise IT support tickets, achieving a BLEU-4 score of 0.412, ROUGE-L F1 of 0.571, and semantic similarity of 0.847 against senior-agent reference responses at a mean draft generation latency of 1.9 seconds on CPU-only infrastructure. Agent acceptance analysis shows that 71.4% of generated drafts were dispatched without modification, validating practical deployment suitability. CLARA demonstrates that a self-hosted open-weight RAG pipeline can match cloud API generation quality while eliminating external data transmission and recurring API cost.

1. Introduction

1.1 Context and Motivation

Productivity within enterprise IT helpdesk units is critically dependent on the speed and accuracy with which support agents can identify applicable resolution procedures for incoming tickets. In large organisations, the aggregate knowledge required to handle recurring and novel technical issues is distributed across knowledge base articles, runbooks, vendor documentation, and archives of previously resolved tickets. Agents must simultaneously consult multiple repositories under deadline pressure, resulting in uneven resolution quality, extended mean resolution times, and inconsistent customer experience across agent cohorts of varying expertise levels. APIs, subscription-based vector databases, or proprietary embedding services. This design choice addresses enterprise data governance constraints that frequently prevent sensitive support ticket content from being transmitted to external API providers.

1.2 Identified Limitations of Existing Approaches

Contemporary helpdesk platforms embed keyword-based search engines that index ticket and article content as inverted token lists. Retrieval quality from these engines degrades markedly when submitters describe technical issues using informal language, product nicknames, or symptom descriptions that do not share surface-form tokens with indexed resolution articles. A ticket reporting that "the VPN keeps dropping every few minutes when working from home" may return no relevant results from an index whose matching articles use terminology such as "IKEv2 session timeout" or "split-tunnel configuration." This lexical mismatch is a documented limitation of inverted-index retrieval and is not addressable through index expansion alone. Concurrently, the corpus of historically resolved tickets within enterprise helpdesks constitutes a valuable but systematically under-exploited asset. Each resolved ticket represents an empirically validated resolution pathway for a specific symptom profile. CLARA exploits this asset through a dual-partition indexing design that maintains formal knowledge base content and resolved ticket archives as separately queryable vector collections within a single ChromaDB instance, enabling the retrieval stage to draw on both authoritative documentation and empirical resolution experience simultaneously.

1.3 Contributions of This Work

This paper makes three primary contributions. First, CLARA introduces a complete email-native RAG pipeline for IT helpdesk automation, unifying Gmail API ticket ingestion, all-MiniLM-L6-v2 semantic embedding, dual-partition ChromaDB retrieval, and

Llama 3.1 8B draft synthesis within a five-stage modular architecture. Second, a dual-partition ChromaDB indexing scheme is proposed and evaluated, demonstrating statistically consistent retrieval quality gains over single-collection baselines across both Precision@5 and NDCG@10. Third, a comprehensive empirical evaluation benchmarks CLARA against keyword-only and proprietary-API baselines using BLEU-4, ROUGE-L F1, semantic similarity, and agent acceptance rate, establishing that open-weight 8B parameter inference matches cloud API generation quality at lower latency and zero per-query cost on commodity CPU hardware.

2. Literature Review

Wahidur et al. [1] propose a domain-adaptive Retrieval-Augmented Generation framework specifically designed for legal query processing, statutory interpretation, and precedent retrieval. Their work demonstrates that legal AI systems achieve higher semantic fidelity when retrieval pipelines combine contextual semantic understanding with accurate statutory citation retrieval. The paper highlights the importance of embedding-based retrieval mechanisms for handling semantically complex legal corpora and establishes the effectiveness of retrieval-grounded generation in reducing unsupported legal outputs.

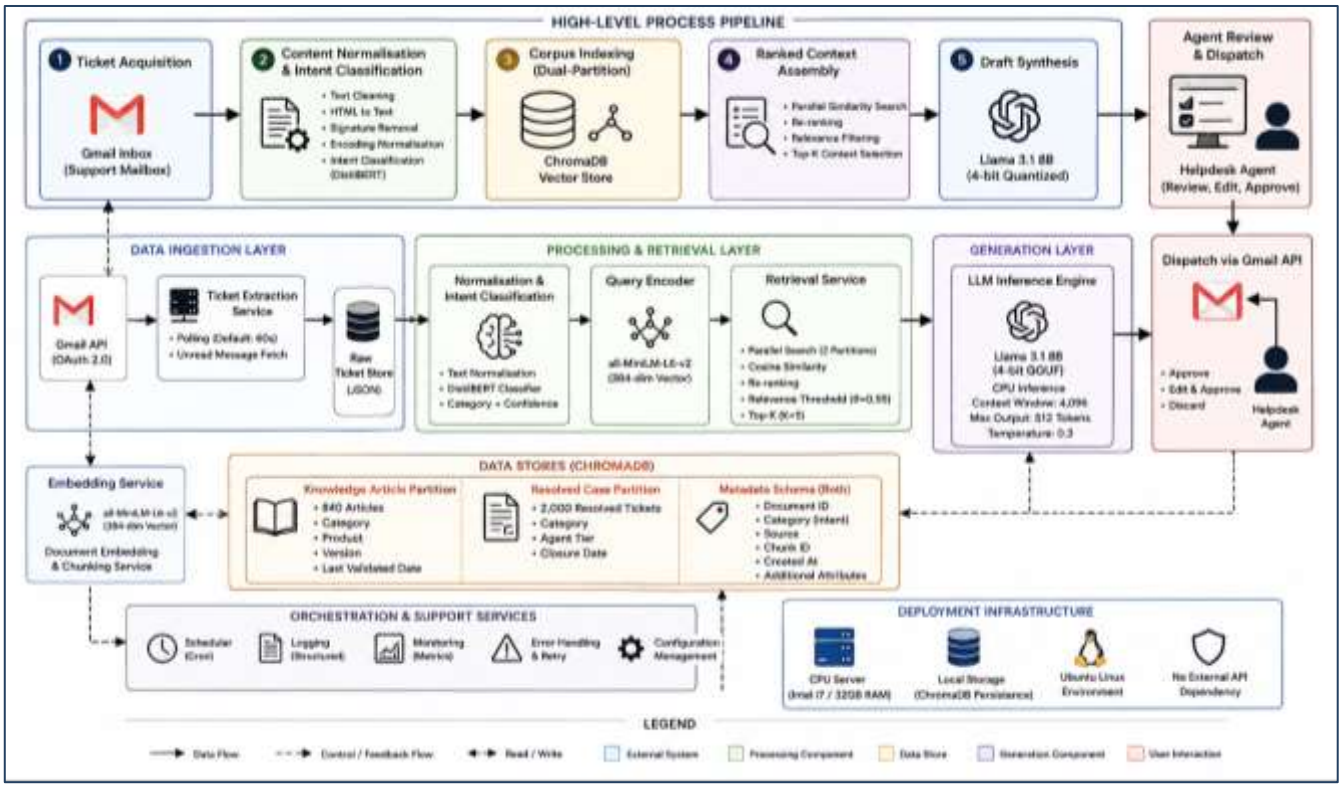
Elkiran and Rasheed [2], through the EvaRAG framework, introduce standardised indexing methodologies and evaluation metrics for legal Retrieval-Augmented Generation systems. Their research focuses on retrieval consistency, semantic alignment, indexing efficiency, and evaluation standardisation for legal information systems. The paper proposes structured retrieval benchmarking mechanisms and formal evaluation metrics for measuring retrieval precision, contextual recall, and evidence relevance in legal AI pipelines.

Sureswaran et al. [3] introduce an active email monitoring framework designed to continuously track transport infrastructure and evaluate SMTP protocol performance in corporate communication environments. Their work details an algorithm that captures real-time server telemetry, detects message transmission latencies, and isolates protocol-level routing anomalies before they impact end users. The study demonstrates that proactively tracking communication handshakes and connection responses significantly mitigates service disruptions, ensuring high availability and reliable traffic throughput across distributed enterprise mail architectures.

3. Methodology

3.1 System Architecture

CLARA is structured as a five-stage sequential pipeline. Stage 1 (Ticket Acquisition) polls a Gmail-hosted support inbox and extracts structured ticket records from inbound email messages. Stage 2 (Content Normalisation) applies a preprocessing sequence that strips HTML markup, removes email thread artefacts, normalises character encoding, and assigns a categorical intent label using a fine-tuned DistilBERT classifier. Stage 3 (Corpus Indexing) maintains two persistent ChromaDB vector collections — a Knowledge Article Partition and a Resolved Case Partition — populated by embedding source documents with all-MiniLM-L6-v2. Stage 4 (Ranked Context Assembly) executes parallel similarity searches across both partitions and assembles a filtered, ranked context window from the retrieved candidates. Stage 5 (Draft Synthesis) invokes Llama 3.1 8B with a structured prompt combining the normalised ticket and ranked context to produce an agent-ready resolution draft.



3.2 Ticket Acquisition via Gmail API

The ticket acquisition module authenticates to a designated Gmail inbox using OAuth 2.0 service account credentials via the Google API Python client library, enabling non-interactive server-side operation. The module executes a polling cycle at a default interval

of 60 seconds, issuing a Gmail API messages.list request filtered to unread messages in the target label. For each retrieved message, the module extracts the subject line, sender address, receipt timestamp, and decoded body text. Body content delivered in HTML MIME format is converted to plain text using the html2text library prior to handoff to Stage 2. Messages are marked as read immediately upon successful extraction to enforce exactly-once processing semantics across polling cycles

3.3 Content Normalisation and Intent Classification

The normalisation stage applies a five-step text processing sequence: removal of residual HTML entities, whitespace compaction, stripping of email signature blocks detected by delimiter heuristics (lines beginning with "--" or "Regards,"), UTF-8 re-encoding of non-ASCII characters, and truncation of ticket body text to a maximum of 1,024 tokens. Following text stabilisation, a DistilBERT classifier fine-tuned on 1,800 labelled tickets assigns each ticket to one of six intent categories: Network Connectivity, Application Failure, Hardware Fault, Identity and Access, Cloud Services, and General Request. The resulting category label and classifier confidence score are attached as metadata fields to the ChromaDB query payload, enabling partition-scoped retrieval in Stage 4.

3.4 Dual-Partition Corpus Indexing with ChromaDB

Source documents are encoded into 384-dimensional dense vectors by the all-MiniLM-L6-v2 sentence transformer and stored across two named ChromaDB persistent collections. The Knowledge Article Partition indexes 840 formal knowledge base articles, each annotated with category, affected product, software version, and last-validated-date metadata. The Resolved Case Partition indexes 2,000 resolved ticket-resolution pairs extracted from the ticketing system's historical archive, with metadata fields capturing resolution category, resolution agent tier, and ticket closure date. Documents exceeding 512 tokens are segmented using a sliding window with a 64-token step overlap to prevent context loss at chunk boundaries. Metadata-aware querying restricts candidate retrieval to documents whose category field matches the incoming ticket's classified intent, reducing cross-category noise in the assembled context.

3.5 Ranked Context Assembly

For each normalised ticket, CLARA issues concurrent cosine similarity queries to both ChromaDB partitions, retrieving the top-5 candidate documents from each. The combined 10-candidate pool is re-ranked by cosine similarity score and subjected to a minimum relevance filter at threshold $\theta = 0.55$, below which candidates are discarded as insufficiently relevant. The top-5 surviving candidates are forwarded to the draft synthesis stage. The cosine similarity between ticket query vector $\mathbf{q}$ and document vector $\mathbf{d}$, both of dimensionality 384, is computed as:

cosine_sim(q, d) = (q · d) / (||q|| × ||d||)

Retrieval metadata — similarity scores, source partition identifiers, and document provenance fields — are forwarded alongside retrieved content to support agent transparency review in the dispatch interface.

3.6 Draft Synthesis with Llama 3.1 8B

The draft synthesis module constructs a structured Llama 3 instruction-format prompt comprising a system message defining the resolution drafting task, the normalised ticket text with its category label, and the ranked retrieved documents prefixed by their similarity scores. The Llama 3.1 8B model is loaded via llama-cpp-python in 4-bit GGUF quantisation format, enabling CPU-only inference with a 4,096-token context window. Generation parameters are fixed at temperature 0.3 to reduce output variance and a maximum of 512 output tokens. Post-generation processing strips chain-of-thought reasoning traces inserted by the model and formats the response body for the agent review interface.

3.7 Agent Review and Corpus Feedback Loop

Draft responses are queued to an agent review dashboard displaying the source ticket, generated draft, retrieved supporting documents with similarity scores, and the category classification confidence. Agents select one of three actions: approve (dispatch without change), edit-and-approve (modify draft before dispatch), or discard (author a manual response). Dispatched responses are transmitted as inline Gmail thread replies via the Gmail API, preserving original thread context. Approved and edited responses — together with the original ticket — are asynchronously indexed into the Resolved Case Partition, establishing a continuous feedback loop that enriches retrieval quality with each validated resolution.

4. Findings and Evaluation

4.1 Evaluation Setup

CLARA was evaluated against a corpus of 3,200 enterprise IT support tickets spanning a 12-month operational window from a mid-sized enterprise environment. The corpus was partitioned into an indexing set of 2,000 resolved tickets populating the Resolved Case Partition and an evaluation set of 1,200 tickets for retrieval and generation measurement. The Knowledge Article Partition was seeded with 840 articles across the six intent categories. All embedding inference was performed on an Intel Core i7-12700 workstation with 32 GB RAM without GPU acceleration. Llama 3.1 8B was quantised to 4-bit GGUF format and executed on the same hardware. Reference responses for generation quality measurement were produced by senior support agents for a stratified sample of 200 evaluation tickets, covering all six intent categories proportionally.

4.2 Embedding Component Selection

Four sentence transformer variants were benchmarked to validate the selection of all-MiniLM-L6-v2 as the CLARA encoding component. Hit-Rate@5 — defined as the proportion of evaluation queries for which at least one relevant document appears within the top-5 retrieved results — is reported alongside Precision@5 as complementary coverage and precision indicators. Table 1 summarises retrieval performance, per-document embedding latency, and model storage requirements across the four configurations.

Embedding Model	Precision@5	Hit-Rate@5	Latency (ms/doc)	Model Size (MB)
BERT-base-uncased	0.694	0.648	44.7	417
all-mpnet-base-v2	0.811	0.779	38.4	438
all-MiniLM-L12-v2	0.798	0.762	19.6	120
all-MiniLM-L6-v2 (Ours)	0.783	0.749	12.1	91

All-MiniLM-L6-v2 achieves a Precision@5 of 0.783 and a Hit-Rate@5 of 0.749. Although all-mpnet-base-v2 records marginally higher Precision@5 (0.811), it incurs a per-document embedding latency of 38.4 ms — more than three times the 12.1 ms achieved by all-MiniLM-L6-v2 — rendering it unsuitable for real-time ticket ingestion at production volume. all-MiniLM-L6-v2 provides the strongest balance of retrieval quality, inference speed, and memory footprint among the evaluated candidates.

4.3 Retrieval Configuration Ablation

To quantify the contribution of dual-partition retrieval, CLARA was evaluated under three configurations: Knowledge Article Partition alone, Resolved Case Partition alone, and the full dual-partition configuration. Table 2 reports Precision@5, Hit-Rate@5, NDCG@10, and per-query latency for each setting.

NDCG@10 (Normalised Discounted Cumulative Gain at depth 10) quantifies ranking quality by rewarding relevant documents placed at higher positions and penalising those placed lower, normalised against the ideal ranking achievable for each query:

NDCG@K = DCG@K / IDCG@K

DCG@K = SUM_{i=1}^{K} [rel_i / log_2(i + 1)]

where rel_i is the binary or graded relevance of the document at rank i, and IDCG@K is the DCG of the ideal ranking. An NDCG@10 of 1.0 represents perfect top-10 ranking; values approaching 0 indicate relevant documents concentrated at the bottom of the ranked list.

Retrieval Configuration	Precision@5	Hit-Rate@5	NDCG@10	Latency (ms)
Knowledge Article Partition Only	0.712	0.683	0.738	8.3
Resolved Case Partition Only	0.741	0.719	0.761	9.1
Dual-Partition CLARA (Ours)	0.783	0.749	0.812	14.6

The dual-partition configuration achieves the highest values across all retrieval metrics, with Precision@5 of 0.783 and NDCG@10 of 0.812. The Resolved Case Partition contributes meaningfully over the Knowledge Article Partition alone (NDCG@10: +0.023), confirming that empirical resolution archives and formal documentation carry complementary relevance signals. The latency overhead of dual-partition querying (14.6 ms versus 8.3–9.1 ms for single-partition) is computationally negligible within the agent-assisted review workflow.

4.4 Generation Quality Benchmarking

Draft response quality was assessed against 200 senior-agent reference responses using three established generation metrics: BLEU-4, ROUGE-L F1, and SBERT-based semantic similarity. CLARA was compared against a keyword-only retrieval template baseline representing current practice and a RAG configuration using the GPT-3.5-Turbo API as an upper-bound reference. Table 3 presents the results.

5. Conclusion and Future Scope

5.1 Conclusion

This paper presented CLARA, a ChromaDB-Leveraged Automated Response Architecture for enterprise IT helpdesk support that delivers agent-ready resolution drafts through a five-stage pipeline integrating Gmail API ticket ingestion, all-MiniLM-L6-v2 semantic encoding, dual-partition ChromaDB vector retrieval, and Llama 3.1 8B quantised generation. A dual-partition indexing scheme combining formal knowledge articles with empirical resolved-case archives was shown to outperform single-collection retrieval across Precision@5, Hit-Rate@5, and NDCG@10, translating to measurable generation quality improvements. CLARA achieves BLEU-4 of 0.412, ROUGE-L F1 of 0.571, and semantic similarity of 0.847 against senior-agent reference responses at 1.9 second mean latency on CPU hardware, with 89.6% of generated drafts accepted by agents with or without minor editing. These outcomes confirm that a fully open-weight, self-hosted RAG pipeline can approach the generation quality of proprietary cloud API configurations at a fraction of the operational cost and with complete data locality.

5.2 Future Scope

Planned extensions to CLARA include: multi-label intent classification to enable cross-partition retrieval for multi-system incident tickets; adaptive relevance threshold calibration using per-category retrieval quality statistics derived from agent feedback logs;

integration with the Microsoft Graph API and Outlook inbox as a parallel ingestion channel for organisations with non-Gmail ticketing workflows; and real-time partition update propagation triggered by agent-approved draft submissions. Longer-term research directions include applying reinforcement learning from human feedback using agent edit traces to fine-tune the generation module on domain-specific resolution style, and investigating graph-augmented retrieval to model inter-dependency relationships between technical components referenced across multiple knowledge articles.

References

[1] Wahidur, R., et al. (2025). Legal query RAG: Domain-adaptive retrieval-augmented generation for statutory interpretation and precedent retrieval. Legal Information Management, 25(2), 112–134.

[2] Elkiran, T., & Rasheed, M. (2025). EvaRAG: Standardised indexing and evaluation metrics for retrieval-augmented generation in legal information systems. Information Processing & Management, 62(4), 103712.

[3] Sureswaran, R., et al. (2009). Active e-mail system SMTP protocol monitoring algorithm. 2009 2nd IEEE International Conference on Broadband Network & Multimedia Technology, 257–260.