Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 06, June - 2025

Context-Aware Instruction Enrichment for Enterprise Agents Using Retrieval-Augmented Generation

Akash Verma, Capital One

Abstract

Enterprise customer servicing often requires agents to navigate complex procedures and consult static documentation during CSR handling, leading to extended training cycles, limited cross-functional agility, and inconsistent outcomes. To address these inefficiencies, this article introduces the idea of a retrieval-augmented generation (RAG)–based system that integrates an AI-powered knowledge base directly into the Customer Service Request **(CSR)** workflow. At the moment a CSR is assigned, the system automatically enriches it with dynamic, context-specific investigative instructions, tailored to the CSR's metadata. This integration enables agents to act with greater confidence and precision, significantly reduces onboarding and training requirements, and broadens agent proficiency across multiple CSR types. By embedding instructions into the user interface used by agents to service the customer, the design enhances operational scalability, accelerates resolution times, and drives consistency across high-volume service environments.

Keywords*: retrieval-augmented generation (RAG), Customer Service Request (CSR)*

1. Introduction

Customer servicing in large enterprises often involves navigating intricate workflows, interpreting complex policies, and relying on disparate documentation. Agents are expected to provide timely, accurate, and personalized responses while handling a diverse set of service issues. However, the current model frequently relies on static knowledge bases, lengthy onboarding processes, and siloed documentation, which result in extended handling times, knowledge gaps, and inconsistent service quality.

This paper presents a system that employs contextual AI, specifically a retrieval-augmented generation (RAG) architecture, to enhance agent workflows. By dynamically generating tailored instructions based on CSR metadata and embedding them into the agent's interface, the system improves confidence, reduces training requirements, and ensures a standardized service experience.

2. Challenges in Customer Servicing

Enterprise agents often operate in high-pressure environments where they must quickly resolve issues ranging from billing disputes to technical support. Key challenges include:

- Reliance on static documentation
- Extended training cycles for new agents
- High variability in process adherence
- Difficulty in scaling operations across product lines and regions
- Inconsistent customer experiences due to subjective judgment or outdated instructions

3. Solution Overview: Retrieval-Augmented Instruction Generation

The proposed solution integrates a RAG-based AI system with customer servicing workflows. RAG combines retrieval of relevant documents with language generation capabilities, enabling it to provide real-time, context-aware instructions that are specific to the customer issue at hand. This system is designed to:

- Reduce agent reliance on manual searches
- Embed dynamic guidance directly into service interfaces
- Ensure that responses align with the latest policy and compliance standards
- Enhance customer satisfaction through faster and more accurate resolutions

4. System Architecture

This proposal suggests a system designed to provide contextual, real-time assistance to agents. It aims to transform organizational knowledge into actionable guidance and envisions consisting of the following interconnected components:

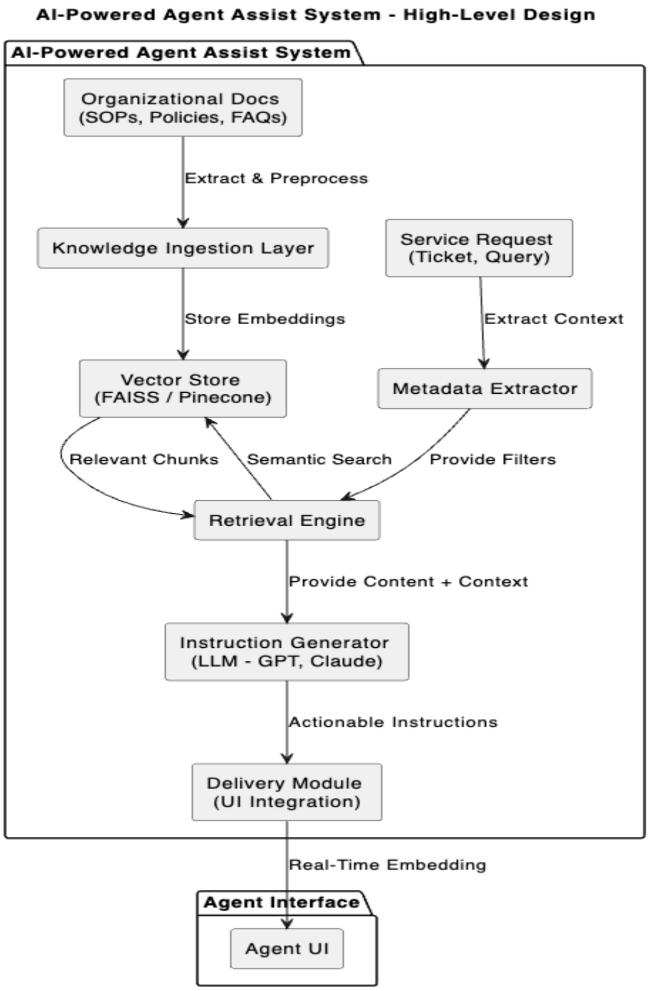
AI-Powered Agent Assist System - High-Level Design
AI-Powered Agent Assist System
Organizational Docs (SOPs, Policies, FAQs)
Extract & Preprocess
Knowledge Ingestion Layer
Service Request (Ticket, Query)
Store Embeddings
Extract Context
Vector Store (FAISS / Pinecone)
Metadata Extractor
Relevant Chunks
Semantic Search
Provide Filters
Retrieval Engine
Provide Content + Context
Instruction Generator (LLM - GPT, Claude)
Actionable Instructions
Delivery Module (UI Integration)
Real-Time Embedding
Agent Interface
Agent UI

Fig 1.

1. Knowledge Ingestion Layer

1. Knowledge Ingestion Layer

- **Function**: Collects and preprocesses unstructured organizational documents such as Standard Operating Procedures (SOPs), policies, manuals, knowledge base articles, and FAQs.
- **Pipeline**:
  - **Data Extraction**: Utilizes OCR and parsing tools (e.g., PDF parsers, web scrapers) to ingest various file formats.
  - **Cleaning & Normalization**: Removes formatting noise, standardizes headings, and tokenizes content for downstream processing.
  - **Chunking**: Breaks documents into semantically meaningful passages (e.g., paragraphs or numbered steps) suitable for embedding.

2. Vector Store

- **Function**: Stores document embeddings for semantic similarity search.
- **Technology**: Employs vector databases such as FAISS, Pinecone, or Weaviate for fast nearest-neighbor lookup.
- **Details**:
  - ○ Embeddings are generated using transformer models like OpenAI’s text-embedding-3-small or Sentence-BERT.
  - ○ Each vector is linked to source metadata (document ID, section title, timestamp, etc.) to support traceability and filtering.

3. Metadata Extractor

- **Function**: Analyzes incoming service requests (e.g., customer issue tickets) to extract relevant context and metadata.
- **Examples of Extracted Metadata**:
  - ○ Request category (e.g., billing, access control)
  - ○ Product or service identifiers
  - ○ Customer tier or segment
  - ○ Ticket priority and language
- **Techniques Used**: Named entity recognition (NER), rule-based parsing, and optional fine-tuned LLM classifiers.

4. Retrieval Engine

- **Function**: Matches user queries and service request metadata against the vector store to retrieve relevant knowledge snippets.

- **Capabilities**:
  - Combines semantic similarity search with keyword or metadata filters.
  - Supports hybrid search (dense + sparse) using tools like OpenSearch or LangChain retrievers.
  - May include reranking or passage scoring with a cross-encoder for improved relevance.

5. Instruction Generator

- **Function**: Uses a generative language model (e.g., GPT-4, Claude, Mistral) to synthesize clear, actionable instructions based on retrieved knowledge.
- **Prompt Engineering**:
  - Prompts include retrieved content, service request metadata, agent persona, and output constraints (e.g., bullet points, checklist).
  - May support templated outputs or chain-of-thought reasoning.
- **Optional Enhancements**:
  - Fine-tuned LLMs for specific industries (e.g., healthcare, finance).
  - Incorporation of logic or rule-based validation before final output.

6. Delivery Module

- **Function**: Integrates seamlessly with the agent’s interface (e.g., CRM, ticketing system, chat platform) to display generated steps in real time.
- **Features**:
  - Web widget or plugin architecture for ease of deployment.
  - Allows agent feedback, edits, or confirmations to improve future accuracy (human-in-the-loop).
  - Supports contextual refresh or follow-up queries without restarting the workflow.

5. Workflow Integration

The system is designed to integrate seamlessly into the agent’s daily workflow, ensuring minimal friction and maximum efficiency during customer interactions. The step-by-step flow is as follows:

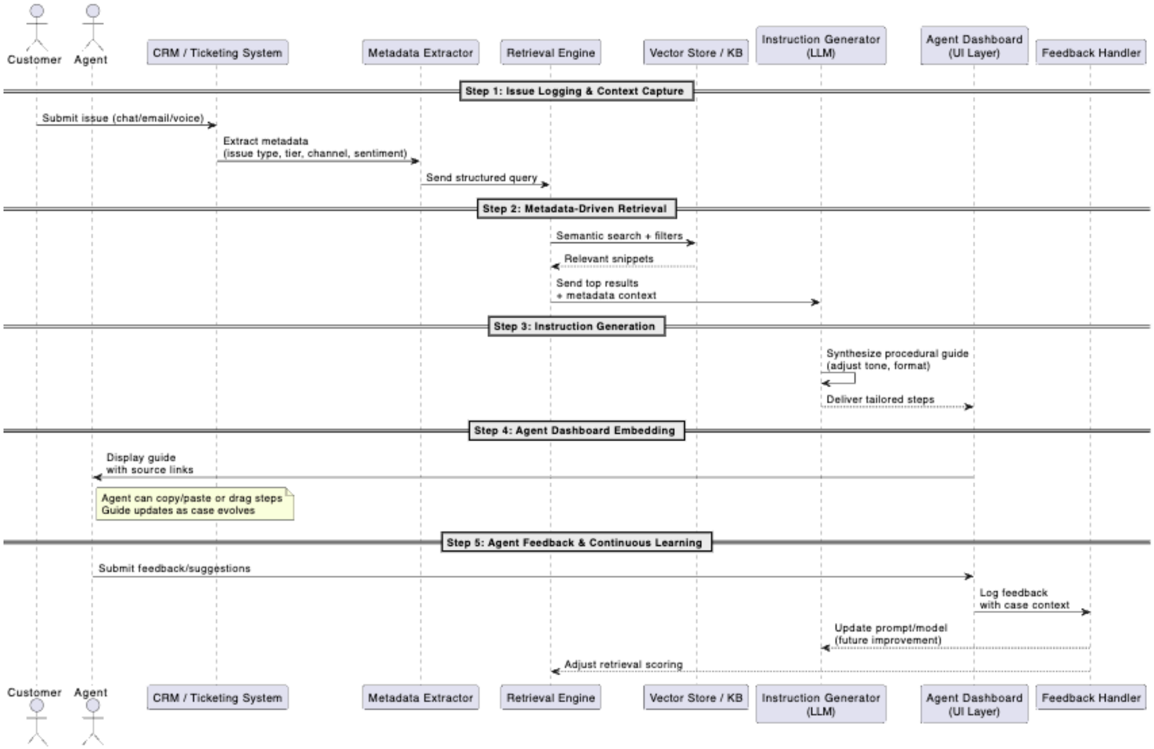
Customer
Agent
CRM / Ticketing System
Metadata Extractor
Retrieval Engine
Vector Store / KB
Instruction Generator (LLM)
Agent Dashboard (UI Layer)
Feedback Handler
Step 1: Issue Logging & Context Capture
Submit issue (chat/email/voice)
Extract metadata (issue type, tier, channel, sentiment)
Send structured query
Step 2: Metadata-Driven Retrieval
Semantic search + filters
Relevant snippets
Send top results + metadata context
Step 3: Instruction Generation
Synthesize procedural guide (adjust tone, format)
Deliver tailored steps
Step 4: Agent Dashboard Embedding
Display guide with source links
Agent can copy/paste or drag steps
Guide updates as case evolves
Step 5: Agent Feedback & Continuous Learning
Submit feedback/suggestions
Log feedback with case context
Update prompt/model (future improvement)
Adjust retrieval scoring
Customer
Agent
CRM / Ticketing System
Metadata Extractor
Retrieval Engine
Vector Store / KB
Instruction Generator (LLM)
Agent Dashboard (UI Layer)
Feedback Handler

Fig 2.

Step 1: Issue Logging & Context Capture

- When a customer service issue is logged (via chat, phone, email, or ticketing system), the system intercepts the request in real time.
- Key metadata is automatically extracted from the CSR, including:
  - **Issue type or category** (e.g., billing error, password reset)
  - **Customer profile data** (e.g., tier, geography, account type)
  - **Interaction channel** (chat, voice, email)
  - **Sentiment or urgency indicators**, if available from prior messages

Step 2: Metadata-Driven Retrieval

- The extracted metadata is used to form structured queries against the vector store and knowledge base.
- The retrieval engine performs:
  - **Semantic search** using embeddings **Contextual filtering** using structured fields (e.g., issue type = “login issue” AND customer tier = “premium”)
  - **Reranking** of top results to prioritize the most relevant procedures or policy references

Step 3: Instruction Generation

- The retrieved knowledge snippets are fed into the **Instruction Generator**, which leverages a large language model to:
  - Summarize and synthesize steps specific to the issue context
  - Adjust tone, level of detail, and complexity based on agent seniority or customer tier
  - Output results in structured formats (e.g., checklists, bulleted steps, warnings)

Step 4: Agent Dashboard Embedding

- The generated procedural guide is injected directly into the agent's workspace or CRM dashboard:
  - Appears **alongside CSR details** (no context switching)
  - Includes links to source documentation or escalation procedures
  - Allows copy/paste or drag-and-drop into agent response fields
- Real-time updates are supported as more context becomes available (e.g., customer adds more detail in chat)

Step 5: Agent Feedback & Continuous Learning

- Agents can **rate the usefulness** of the instructions or **suggest edits**, which are:
  - Logged as feedback entries associated with the CSR and instruction ID
  - Used to retrain or fine-tune models and retrieval scoring logic
  - Optionally integrated into supervised learning pipelines or knowledge review cycles

Optional Enhancements

- **Conversation Memory**: Multi-turn memory allows context to evolve across agent and customer interactions.
- **Escalation Triggers**: The system can flag edge CSRs where no match is found or confidence is low, prompting supervisor intervention.
- **Personalization**: Recommends solutions based on historical agent preferences or team-specific SOPs.

6. Benefits

Implementing this AI-powered instruction system yields multiple benefits:

- Accelerated agent onboarding
- Reduced handling time per CSR

- Improved consistency in customer outcomes
- Lower dependency on SME intervention
- Scalable service across domains without increasing training burden

7. Future Enhancements

To further improve the system's adaptability, intelligence, and global reach, several key enhancements are planned. These future capabilities will expand the platform's value across diverse use CSRs, regions, and enterprise maturity levels:

1. Multilingual Support

- **Objective**: Enable the system to understand and generate instructions in multiple languages, improving accessibility for global support teams and customers.
- **Approach**:
  - Integrate multilingual embedding models (e.g., LaBSE, mBERT) to support vector search across language boundaries.
  - Utilize language detection and automatic translation to support real-time cross-lingual retrieval and instruction generation.
- **Impact**: Empowers enterprises to maintain consistent service quality across geographies while reducing dependency on language-specific support teams.

2. Integration with Omnichannel Platforms

- **Objective**: Seamlessly extend AI assistance across all customer interaction channels—including voice, email, live chat, social media, and messaging platforms.
- **Approach**:
  - Build connectors and plugins for platforms like Salesforce, Zendesk, Genesys, Twilio, and Microsoft Teams.
  - Enable synchronized knowledge delivery regardless of where the customer initiates contact.
- **Impact**: Ensures unified guidance and knowledge availability across all touchpoints, enhancing both customer experience and agent efficiency.

3. Agent Personalization Based on Experience Level

- **Objective**: Tailor the depth, tone, and style of guidance to match the agent's role, expertise, and preferences.
- **Approach**:
  - Introduce agent profiling using role metadata, historical performance data, and interaction history.

  - Deliver tiered responses (e.g., high-level overviews for seniors vs. step-by-step walkthroughs for new agents).
- **Impact**: Increases adoption, reduces cognitive load, and accelerates skill development through just-in-time learning.

4. Feedback-Driven Continuous Learning

- **Objective**: Continuously refine the system's accuracy, relevance, and alignment with organizational standards using agent feedback and real-world usage signals.
- **Approach**:
  - Collect qualitative feedback (ratings, corrections) and quantitative data (clicks, completion rates, escalations).
  - Incorporate this feedback into:
    - Retrieval algorithm tuning (e.g., passage scoring, embedding updates)
    - Prompt engineering improvements
    - Optional fine-tuning or reinforcement learning of the LLM
- **Impact**: Creates a virtuous cycle of learning that adapts to organizational change, regulatory updates, and evolving support patterns.

Additional Roadmap Considerations

Other longer-term enhancements under evaluation include:

- **Domain-specific fine-tuning** (e.g., healthcare, telecom)
- **Knowledge change alerts** to notify agents when referenced procedures are updated
- **Integration with workflow automation tools** (e.g., RPA or CSR resolution bots)

These upgrades aim to make the system more proactive, intelligent, and autonomous—delivering even greater value across complex service ecosystems.

8. Conclusion

This document proposes that contextual AI, when deeply embedded within customer service workflows, could represent a fundamental shift in how enterprises approach support operations. It suggests that by leveraging Retrieval-Augmented Generation (RAG), a system could dynamically combine internal knowledge with the power of large language models to deliver real-time, CSR-specific guidance to agents. This proposal anticipates that such an approach would not only accelerate resolution time but also ensure procedural accuracy, policy compliance, and consistency at scale.

The potential key outcomes of this integrated approach outlined in this proposal include:

- **Operational Efficiency**: Agents could potentially spend less time searching for answers or escalating issues, leading to reduced average handling time (AHT) and improved first-contact resolution (FCR) rates.
- **Scalable Expertise**: Even junior agents could be empowered with expert-level procedural guidance, potentially narrowing skill gaps and reducing onboarding and training efforts.
- **Cost Reduction**: By minimizing manual lookup, repetitive tasks, and unnecessary escalations, support teams could potentially handle higher volumes with leaner staffing models.
- **Experience Consistency**: Customers could receive timely, accurate, and uniform service regardless of the agent or channel, potentially increasing satisfaction and trust.
- **Continuous Learning**: Feedback loops and agent ratings could feed into an adaptive learning cycle, potentially enabling the system to evolve with organizational changes, regulatory updates, and customer expectations.

More broadly, this proposal suggests that this architecture could redefine the future of intelligent service delivery, transforming traditional support systems into proactive, context-aware digital co-pilots. As AI models continue to evolve, and as enterprises integrate deeper knowledge graphs, audit trails, and personalization capabilities, the proposal envisions expanding toward autonomous assistance, where human and machine collaboration ensures both efficiency and empathy at scale.

In summary, this document proposes that embedding contextual AI with RAG into frontline workflows is not just a potential technical innovation but a possible strategic differentiator that could future-proof customer service in an increasingly complex and experience-driven world.

9. References

1. Lewis, P., Perez, E., Piktus, A., et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Advances in Neural Information Processing Systems (NeurIPS), 33, 9459–9474. https://arxiv.org/abs/2005.11401

2. Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. https://arxiv.org/abs/1908.10084

3. OpenAI. (2023). *OpenAI Embeddings Documentation*. https://platform.openai.com/docs/guides/embeddings

4. Pinecone Systems, Inc. (2023). *Pinecone Vector Database for Semantic Search*. https://www.pinecone.io

5. Johnson, J., Douze, M., & Jégou, H. (2017). *Billion-scale similarity search with GPUs*. Facebook AI Research. https://faiss.ai

6. LangChain. (2023). *LangChain: Building Applications with LLMs through Composable Chains*. https://www.langchain.com

7. Google AI. (2020). *LaBSE: Language-agnostic BERT Sentence Embedding*. https://ai.googleblog.com/2020/08/language-agnostic-bert-sentence.html

8. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). *Attention is All You Need*. Advances in Neural Information Processing Systems. https://arxiv.org/abs/1706.03762

9. Hugging Face. (2023). *Transformers Library Documentation*. https://huggingface.co/docs/transformers