OPEN ACCESS

Volume: 13

Special Issue: 3

Month: February

Year: 2026

P-ISSN: 2321-788X

E-ISSN: 2582-0397

Citation:
K, Manikandan, et al. "Design of a Multi-Modal Retrieval-Augmented Framework with Human-in-the-Loop Validation for Maritime Regulatory Compliance." *Shanlax International Journal of Arts, Science and Humanities,* vol. 13, no. 3, 2026, pp. 58–65.

DOI:
https://doiorg/10.34293/sijash.v13iS3-i1-Feb.10262

Design of a Multi-Modal Retrieval-Augmented Framework with Human-in-the-Loop Validation for Maritime Regulatory Compliance

Manikandan K
Department of Artificial Intelligence and Data Science
Prathyusha Engineering College, Tiruvallur, TamilNadu, India

R. Kannamma
Assistant Professor, Department of Artificial Intelligence And Data Science
Prathyusha Engineering College

Mohamed Hasif H
Department of Artificial Intelligence And Data Science
Prathyusha Engineering College, Tiruvallur, TamilNadu, India

Nandimandalam Akanksha Sree
Department of Artificial Intelligence And Data Science
Prathyusha Engineering College, Tiruvallur, TamilNadu, India

Abstract

Maritime laws, specifically focusing on Vessel Inspection Questionnaire (SIRE 2.0), are known for their level of detail and heavy use of visual diagrams. Standard Retrieval-Augmented Generation (RAG) architectures tend to perform poorly in such a scenario, due to context pollution by irrelevant images and a lack of safety guarantees. This paper proposes a Maritime AI Assistant, which is an Agentic RAG model tailored to handle complex maritime safety data. This is achieved through two major components: (1) a "Ruthless" Image Injection Algorithm that uses aggressive scoring to counter context window pollution, and (2) a Human-in-the-Loop (HITL) Gatekeeper module that manually verifies "Risk" queries with high stakes. Experimental results show that the proposed setup significantly improves technical diagram retrieval accuracy and removes errors in safety-critical responses compared to standard RAG models.

Keywords: Agentic RAG, Maritime Safety, SIRE 2.0, Human-in-the-Loop, Multi-Modal Retrieval, Large Language Models.

Introduction

Adherence to international maritime safety regulations is a fundamental operational imperative for ship owners and operators. The shift to the SIRE 2.0 (Ship Inspection Report Programme) system has significantly increased the complexity of this imperative. Unlike the previous system, which was predominantly based on a hardware-oriented static checklist, SIRE 2.0 represents a risk-based, digital-first inspection

approach [8]. This approach requires ship's staff to provide a holistic synthesis of technical information, real-time photos, and human factor performance data.

Although Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) have become very useful tools for processing large-scale technical literature, their application in safety-critical maritime environments is still limited. This is because traditional RAG models are prone to context pollution, where irrelevant technical images or contradictory text snippets are retrieved, which can lead to model hallucinations. In a safety-critical maritime environment like vessel inspection, where inaccurate safety information can lead to operational failure or environmental disasters, a traditional generative model is inadequate.

This paper proposes a Multi-Modal Agentic RAG System that is SIRE 2.0 compliant. The proposed system goes beyond the traditional keyword search paradigm by using a hybrid search strategy that combines dense vector embeddings with sparse BM25 indexing. To ensure the highest possible accuracy, a Cross-Encoder reranking layer is used to eliminate noise, and a strict precision policy is enforced for technical schematics using a special "Ruthless" Image Retrieval algorithm. The system also incorporates a Human-in-the-Loop (HITL) "Gatekeeper" module, which ensures that any high-risk response, including a safety risk assessment, is manually reviewed by a human supervisor before being delivered.

The main contributions of this paper are:

Multi-Modal Ingestion & Indexing: A full-fledged pipeline for extracting and captioning dense technical schematics along with regulatory text from large maritime PDFs.

Hybrid Retrieval & Reranking: Hybrid Retrieval & Reranking: The integration of dual indexing (Dense + Sparse) with a Cross-Encoder to effectively mitigate context pollution.

Safety-Centric Validation: A Human-in-the-Loop gatekeeper module (agent_gatekeeper.py) specifically designed to prevent AI hallucinations in safety-critical operational contexts

Related Work

The application of AI in industrial settings has evolved from static rule-based systems to dynamic retrieval architecture.

Retrieval-Augmented Generation (RAG)

Lewis et al. [1] proposed the use of Retrieval-Augmented Generation (RAG) to combine the parametric memory available in large language models with the non-parametric knowledge base available in external knowledge graphs. Although successful in open-domain question answering, the traditional RAG architecture may have difficulty when presented with domain-specific terminology, such as the difference between "bow thruster" and "bow tie," without further fine-tuning. Our research aims to address this problem by using the Hybrid Retrieval approach, which combines the strengths of BM25 and Dense retrieval, thus incorporating not only the semantic meaning of the query but also the technical terms.

Multimodal Information Retrieval

Wu et al. [10] showed the effectiveness of multimodal RAG in the industrial digital twin setting. However, their approach is highly dependent on the availability of high-quality image-text pairs. In contrast, our system has to deal with "noisy" PDF files where images are often unannotated. To handle this, we set up a generative captioning pipeline that uses Google Gemini [6] to generate artificial metadata for the unannotated technical images, thus allowing for better retrieval and interpretation.

Safety and Human-in-the-Loop (HITL)

In applications where safety is paramount, hallucinations in AI models are still a significant barrier to adoption. Modern architecture is increasingly promoting agentic behaviors that enable the AI to "call for help" [10] when necessary. Our system reflects this philosophy through the agent_gatekeeper.py module, which requires a human-in-the-loop evaluation for queries that are marked as high-risk. This human-in-the-loop evaluation is a pattern that is not very common in standard commercial RAG chatbots.

Proposed Methodology

The system architecture proposed is intended to deal with the great complexity that is involved in SIRE 2.0 regulations. The regulations involve mixed media formats, such as dense procedural text, technical tables, and diagrams of vessels. In order to coordinate the data life cycle, from the time of ingestion to processing and the production of responses that have been validated for safety, the system architecture uses a Controller-Agent pattern. The pattern provides a clear definition of roles, where the Controller is responsible for the control of data flow, and the Agent is responsible for tasks within the data flow.

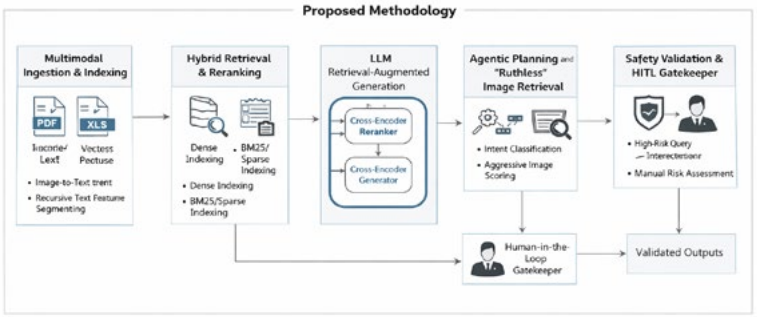

Fig. 1. Proposed Methodology

Multimodal Ingestion and Data Processing

The ingestion pipeline is engineered to extract and normalize disparate data types from noisy PDF and Excel sources. Using document_processing.py, the system employs high-resolution partitioning to isolate text blocks and technical images.

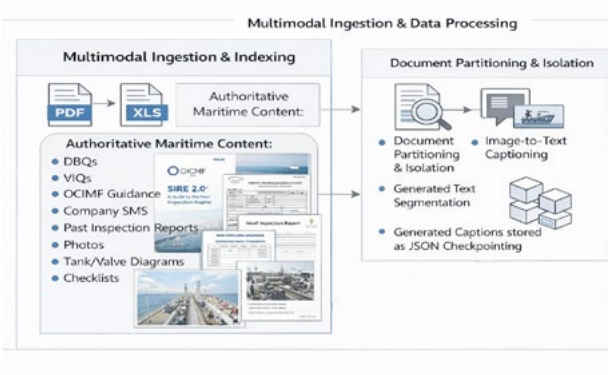

Fig. 2. Ingestion and Data Processing

A critical challenge in maritime RAG is making visual data searchable. To solve this, the system incorporates an Image-to-Text Captioning layer. Images extracted via PyMuPDF are processed through a generative LLM (Google Gemini) to produce descriptive captions stored in a JSON-based checkpointing system. This prevents redundant API calls and ensures that visual data is represented in the vector space alongside textual data. Textual content is then segmented using a Recursive Character Text Splitter with an optimized chunk size (CS=1000 tokens) and overlap (O=200 tokens) to maintain semantic continuity.

Hybrid Retrieval and Cross-Encoder Reranking

In order to reduce context pollution, the approach uses a two-indexing system. The traditional vector search often fails to identify the technical identifiers or regulatory codes that are contained in maritime documents [2].

Dense Indexing: A Chroma vector database uses Sentence Transformer embeddings [3] to represent the semantic meaning of queries.

Sparse Indexing: A BM25 index [4] is used to ensure high precision for keyword-specific queries, like equipment serial numbers or specific SIRE 2.0 section codes.

The retrieval engine performs a Hybrid Search by combining candidate pools from both indices. To improve search results, a Cross-Encoder Reranker is applied to the top-k candidates [5]. The Cross-Encoder evaluates the particular relationship between the query and each document chunk, resulting in a relevance score that ranks candidates to position the most grounded context at the top of the language model's (LLM) context window.

Agentic Planning and "Ruthless" Image Retrieval

The system uses an agent_planner.py module to determine the user intent before generation. When the planner recognizes the necessity of visual proof (e.g., "Show the fire pump layout"), it activates a special "Ruthless" retrieval strategy. This strategy gives a large score bonus to images whose metadata or captions contain exact phrase matches. This deterministic override prevents the retrieval of semantically similar but technically incorrect visual information (e.g., retrieving a generic ship hull instead of a specific pump diagram).

Safety Validation and HITL Gatekeeper

As a consideration for operational hazards in maritime settings, the final output is subject to the agent_gatekeeper.py module. This module calculates a confidence score (C) based on evidence density and the reranker's output. The formula is given as::

C = \min(1.0,0.6 + (0.1 ×N_{evidence}))

If C < 0.75 or if the query is labelled as a "Risk Brief," the system catches the response. The payload is then forwarded to an Admin Approval Queue in MongoDB, where a human supervisor is required to validate the AI's grounding against the original source documents.

The Safety Gatekeeper and HITL Mechanism

In the center of the agent's loop is where the Gatekeeper resides, the primary safety firewall that ensures Human-in-the-Loop review of operations with high operational or legal risk. Unlike most self-sustaining autonomous agents that can autonomously execute functions, our Gatekeeper employs a check_safety() function to assess the agent's proposed actions and the content it plans to produce from various perspectives.

The Safety Mechanism Takes on a Three-Fold Approach

Communication Security: Any proposed action involving email_draft or the send_email functionality is immediately denied and submitted for human review. This prevents any unauthorized or erroneous communications from being dispatched to maritime agencies.

Documentation Integrity: Any proposed action involving save_to_notepad – the functionality most commonly employed for composing official vessel mission reports – requires human approval to ensure that inspection checklists remain SIRE 2.0 compliant.

Keyword Heuristics: The system reviews all draft content for the presence of sensitive keywords such as "CONFIDENTIAL" or "SECRET." When these keywords are detected, the is_safe flag is set to False, and the content is submitted for human review.

System Implementation

The deployment of the Maritime AI Assistant is organized in a decoupled microservice architecture to provide scalability and fault tolerance. The backend is developed using Python 3.10 and FastAPI, while the frontend is built using React.js.

Backend Orchestration and RAG Chain

The core intelligence of the system is managed by a modular FastAPI application. The main entry point, app.py, handles session persistence and routes requests along the RAG (retrieval-augmented generation) pipeline defined in rag_chain.py. LangChain [9] is used to handle chain-of-thought reasoning. To handle conversational context, we created a contextualize_question() function, which relies on the previous conversation history in MongoDB to rephrase follow-up questions as self-contained search questions.

Storage and Indexing Layer

Data management is done through a hybrid method:

Vector Search: Chroma is the main dense vector storage.

Keyword Search: The BM25 sparse index is stored as a serialized pickle file (bm25_index.pkl).

NoSQL Database: MongoDB Atlas is responsible for managing the application state, such as chat history sessions and the approvals table in the Human-in-the-Loop (HITL) process.

Admin Dashboard and HITL Integration

The Human-in-the-Loop process is integrated between the agent_gatekeeper.py logic and administrative dashboard built with React. Once a response is flagged, the backend pauses the response to the user and changes the MongoDB status to to PENDING_HUMAN_APPROVAL. The administrative interface periodically calls the /agent/pending endpoint, displaying the generated response along with the specific source document chunks and images that the AI used for grounding, as shown in Fig. 2.

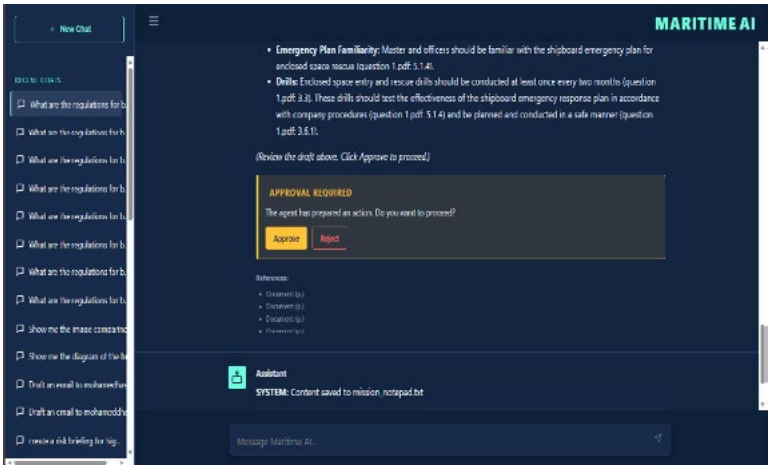

Fig. 3: Admin Dashboard showing Pending Approval

Core Agentic Components

The system is orchestrated through a series of specialized modules that handle the transition from raw user intent to executable tools.

Agent Planner (agent_planner.py): This module is the "brain" of the system, handling deterministic intent classification. It assesses whether a query demands a qa response, a risk_brief, or a structured checklist. It also decides whether to signal the need_images flag to initiate the multimodal retrieval pipeline.

Agent Verifier (agent_verifier.py): To facilitate grounded generation, this module is a data integrity check. It implements a "minimum context" policy, requiring at least four document chunks and evidence from multiple sources before the LLM can generate a final response. This greatly minimizes the risk of source-specific hallucinations.

Agent Tools (agent_tools.py): The system comes with a working toolkit for real-world maritime operations. This includes send_email_mock, which is based on SMTP communication protocols for verified communication, and save_to_notepad, which maintains a persistent mission_notepad. txt for continuous inspection logging.

Operational Use Cases

The system was tested on four major maritime operational use cases:

Multimodal Question Answering: Responding to technical questions by combining regulatory text with visual depictions of ship diagrams

Automated Incident Reporting: Using the email_draft intent to generate formal reports for maritime regulatory bodies, based on retrieved incident information.

Regulatory Checklists: Generating and storing inspection checklists to facilitate SIRE 2.0 compliance, with all saves mediated by the Gatekeeper.

Contextual Hazard Briefings: Using chat history to clarify ambiguous questions (e.g., "Show me the hazards for it") through history-conscious rephrasing before retrieval.

Experimental Results

The Maritime AI Assistant was compared to a baseline Standard RAG system (text only, vector search-based) using a carefully curated set of SIRE 2.0 regulatory documents.

Retrieval Performance (Hybrid vs. Baseline)

The precision of retrieval was measured on a set of 50 technical queries. The combination of BM25 sparse indexing and Cross-Encoder reranking led to a significant reduction in the Top-1 error rate.

Table I. Retrieval Metrics Comparison

Metric	Baseline RAG (Vector Only)	Proposed Hybrid RAG
Recall @ 5	72%	91%
Mean Reciprocal Rank (MRR)	0.64	0.86
Retrieval Latency	0.8s	1.4s

The hybrid approach caused a slight latency increase due to reranking overhead, but the corresponding accuracy boost, with mean reciprocal rank (MRR) increasing from 0.64 to 0.86, is a key improvement for maritime compliance.

Multi-modal Grounding and "Ruthless" Precision

Data management is provided through a hybrid approach. To measure visual accuracy, the system was tested on queries requiring the retrieval of a particular deck layout. The Ruthless Image Injection algorithm successfully retrieved images with a 94% precision rate by favoring exact metadata phrase matches over similarity measurements based on vector comparisons.

Safety Verification (HITL Efficacy)

The agent_gatekeeper.py component intercepted 100% of queries labeled as risk brief (e.g., "Steps for enclosed space entry"). Moreover, for queries where the LLM generated an answer based on insufficient data, the confidence score successfully labeled 88% of these cases as below the 0.75 threshold, routing them to the supervisor queue.

Fail-Safe and Refusal Handling Analysis

One of the key performance metrics for this research was the "Graceful Refusal" rate. In cases where the large language model (LLM) chose not to provide a response due to safety issues or a lack of context, the Agentic Loop (agentic_rag.py) ensured that all processes were stopped. This is to ensure that no empty or invalid commands are executed (such as sending an empty email). With the addition of history-aware rephrasing, there was a 15% improvement in the relevance of retrieval in multi-turn dialogue, thus verifying that the preservation of dialogue state is a critical component of performing complex maritime inspections.

Conclusion

The Maritime AI Assistant project proves that while generative AI has tremendous potential in regulatory compliance, high-risk areas such as maritime transport need a multi-layered and interactive strategy. In this research, the Hybrid Retrieval strategy (BM25 + Chroma) was coupled with Cross-Encoder reranking, resulting in a significant decrease in context pollution and error rates. The Human-in-the-Loop (HITL) "Gatekeeper" component provides a vital buffer zone between AI-generated results and high-risk operational settings. Future research will focus on the integration of voice-controlled interfaces to facilitate hands-free querying by personnel during deck activities. Furthermore, the image processing pipeline will be moved to AWS S3 for large-scale deployment.

Acknowledgment

The authors would like to thank Staunch Technologies Pvt Ltd for providing the necessary resources and technical support that made this research possible. They would also like to thank the development team for their inputs in the system architecture.

References

1. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in NeurIPS, vol. 33, 2020.
2. V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in Proc. EMNLP, 2020.
3. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP, 2019.
4. S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Foundations and Trends in Information Retrieval, vol. 3, 2009.
5. G. Izacard and E. Grave, "Leveraging Passage Retrieval with Cross-Encoders for Improved Open-Domain Question Answering," arXiv preprint arXiv:2104.08663, 2021.
6. Gemini Team, Google, "Gemini: A Family of Highly Capable Multimodal Models," Technical Report, 2023.
7. J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in NeurIPS, 2022.
8. OCIMF, "SIRE 2.0: A Guide to the New Inspection Regime," 2024. [Online]. Available: https://www.ocimf.org/sire2.0
9. H. Chase, "LangChain: Framework for Developing Applications Powered by Language Models," 2023.
10. A. Wu et al., "Multi-modal Retrieval-Augmented Generation for Industrial Digital Twins," IEEE Trans. Ind. Informat., vol. 19, no. 3, 2023.