



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών «Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Εξειδικευμένου Μεγάλου Γλωσσικού Μοντέλου για τη Ναυτιλία με Αξιοποίηση Αρχιτεκτονικής RAG Developing a Maritime-Specific LLM utilizing RAG Architectures
Ονοματεπώνυμο Φοιτητή	Σαμπάνη Τριανταφυλλιά
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΣΠ2339
Επιβλέπων	Διονύσιος Σωτηρόπουλος, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Σεπτέμβριος 2026**

3-Member Examination Committee

Τριμελής Εξεταστική Επιτροπή

D. Sotiropoulos

Associate Professor

Διονύσιος Σωτηρόπουλος
Αναπληρωτής Καθηγητής

G. Tsihrintzis

Professor

Γεώργιος Τσιχριντζής
Καθηγητής

E. Sakkopoulos

Professor

Ευάγγελος Σακκόπουλος
Καθηγητής

Abstract

Maritime technical manuals contain extensive operational, maintenance, and safety-related information, making the retrieval of specific technical details a time-consuming task. This thesis examines the use of Retrieval-Augmented Generation (RAG) as a method for improving access to information contained in maritime technical documentation. A locally deployed RAG system was designed and implemented for question answering over maritime technical manuals. The system processes PDF documents into overlapping text chunks, converts them into vector representations using the all-MiniLM-L6-v2 SentenceTransformer model, and stores them in a persistent ChromaDB vector database. For each user question, relevant document chunks are retrieved and used as context for the Qwen2.5-3B-Instruct language model. Both retrieval and response generation are performed locally, without relying on an external language-model API, while a graphical user interface was developed to support direct interaction with the system. The system was evaluated using 24 questions based on four maritime technical manuals. Relevant information was successfully retrieved for 20 of the 24 questions, corresponding to an overall retrieval success rate of 83.3%. Of the generated responses, 15 (62.5%) were classified as Fully Correct, six (25.0%) as Partially Correct, and three (12.5%) as Incorrect. The results show that the proposed lightweight local RAG system can support access to technical information across different maritime manuals, while also highlighting that successful retrieval does not always guarantee a complete or accurate final answer. The system therefore provides a useful basis for further development of locally deployed RAG tools for maritime technical documentation.

Περίληψη

Τα ναυτιλιακά τεχνικά εγχειρίδια περιλαμβάνουν εκτεταμένες πληροφορίες που σχετίζονται με τη λειτουργία, τη συντήρηση και την ασφάλεια, γεγονός που καθιστά την αναζήτηση συγκεκριμένων τεχνικών στοιχείων χρονοβόρα. Η παρούσα διπλωματική εργασία εξετάζει τη χρήση της αρχιτεκτονικής Retrieval-Augmented Generation (RAG) ως μέθοδο για τη βελτίωση της πρόσβασης σε πληροφορίες που περιέχονται στη ναυτιλιακή τεχνική τεκμηρίωση. Σχεδιάστηκε και υλοποιήθηκε ένα τοπικά εκτελούμενο σύστημα RAG για την παραγωγή απαντήσεων βάσει ναυτιλιακών τεχνικών εγχειριδίων. Το σύστημα επεξεργάζεται έγγραφα PDF σε επικαλυπτόμενα τμήματα κειμένου, τα μετατρέπει σε διανυσματικές αναπαραστάσεις χρησιμοποιώντας το μοντέλο all-MiniLM-L6-v2 SentenceTransformer και τα αποθηκεύει σε μια μόνιμη διανυσματική βάση δεδομένων ChromaDB. Για κάθε ερώτηση του χρήστη, ανακτώνται τα σχετικά τμήματα των εγγράφων και χρησιμοποιούνται ως πλαίσιο για το γλωσσικό μοντέλο Qwen2.5-3B-Instruct. Τόσο η ανάκτηση όσο και η παραγωγή απαντήσεων πραγματοποιούνται τοπικά, χωρίς τη χρήση εξωτερικού API γλωσσικού μοντέλου, ενώ αναπτύχθηκε και γραφικό περιβάλλον χρήστη για την άμεση αλληλεπίδραση με το σύστημα. Το σύστημα αξιολογήθηκε με 24 ερωτήσεις βασισμένες σε τέσσερα ναυτιλιακά τεχνικά εγχειρίδια. Η σχετική πληροφορία ανακτήθηκε επιτυχώς σε 20 από τις 24 ερωτήσεις, που αντιστοιχεί σε συνολικό ποσοστό επιτυχούς ανάκτησης 83,3%. Από τις παραγόμενες απαντήσεις, 15 (62,5%) χαρακτηρίστηκαν ως πλήρως ορθές, έξι (25,0%) ως μερικώς ορθές και τρεις (12,5%) ως λανθασμένες. Τα αποτελέσματα δείχνουν ότι το προτεινόμενο ελαφρύ και τοπικά εκτελούμενο σύστημα RAG μπορεί να υποστηρίξει την πρόσβαση σε τεχνική πληροφορία από διαφορετικά ναυτιλιακά εγχειρίδια, ενώ παράλληλα αναδεικνύουν ότι η επιτυχής ανάκτηση δεν εγγυάται πάντοτε μια πλήρη ή ακριβή τελική απάντηση. Το σύστημα παρέχει επομένως μια χρήσιμη βάση για την περαιτέρω ανάπτυξη τοπικά εκτελούμενων εργαλείων RAG για ναυτιλιακή τεχνική τεκμηρίωση.

Table of Contents

Abstract	2
Περίληψη	2
Chapter 1 – Introduction	5
1.1 Large Language Models	5
1.2 Limitations of Large Language Models	6
1.3 Retrieval-Augmented Generation (RAG)	6
1.4 Challenges of Technical Information Retrieval in the Maritime Industry	7
1.5 Aim and Contribution of the Thesis	14
Chapter 2 – System Design and Implementation	15
2.1 Overall System Architecture	15
2.2 Document Processing and Indexing	16
2.3 Query Processing and Retrieval	17
2.4 Response Generation	17
Chapter 3 – Experimental Evaluation	23
3.1 Evaluation Methodology	23
3.4.1 Retrieval Evaluation	25
3.4.2 Answer Quality Evaluation	26
3.5 Experimental Results	26
3.5.1 Generator Manual Results	27
3.5.2 ECDIS Manual Results	28
3.5.3 MFC Valve Manual Results	30
3.5.4 Boiler Manual Results	31
3.5.5 Overall Results	33
3.6 Error Analysis and Failure Cases	34
3.7 Discussion	36
Chapter 4 – Conclusion and Future Work	39
4.1 Summary of the Work	39
4.2 Main Findings	39
4.3 Limitations	40
4.4 Future Work	41
Chapter 5 – Bibliography	43

Appendix A – Evaluation Questions and Reference Information	45
A.1 Generator Manual Evaluation Questions	45
A.2 ECDIS Manual Evaluation Questions	46
A.3 MFC Valve Manual Evaluation Questions	47
A.4 Boiler Manual Evaluation Questions	48

Chapter 1 – Introduction

1.1 Large Language Models

Large Language Models (LLMs) are deep learning models designed to process and generate natural language. They are trained on large collections of textual data, and thus learn statistical and semantic relationships between words and longer textual sequences. As a result, LLMs can perform a wide range of natural-language processing tasks, including text generation, summarization, translation, question answering, and information extraction [18]. Rather than being explicitly programmed for each individual task, large language models learn general language representations during training and can subsequently be adapted or instructed to perform different tasks [18].

The development of LLMs is closely associated with the Transformer architecture introduced by Vaswani et al. [1]. One of its main differences from earlier sequential architectures is the use of attention mechanisms. These mechanisms allow the model to capture relationships between different elements of the input. In self-attention, tokens at different positions within the same sequence can interact with each other [1]. The model can therefore identify which tokens are more relevant when processing a particular token. As a result, each token is represented according to the context provided by the surrounding sequence. This ability to capture contextual relationships efficiently has contributed to the widespread use of Transformers in modern language models. This architecture also forms the basis of the Transformer-based models used in the implementation of the proposed RAG system, which are presented in the following sections.

Large language models are typically trained on extensive text collections in order to acquire general linguistic patterns and knowledge [18]. During this initial pre-training stage, the model learns statistical relationships and patterns within language from large amounts of textual data. However, a pre-trained model is not necessarily optimized to follow user instructions or perform specific tasks. Models may therefore undergo additional adaptation or instruction tuning [19]. During instruction tuning, the model is further trained on tasks expressed through natural-language instructions and corresponding responses. This process improves the model's ability to follow instructions and has been shown to improve its generalization to unseen tasks [19].

In the present thesis, Qwen2.5-3B-Instruct is used as the language-generation component of the proposed RAG system. Qwen2.5 is a Transformer-based family of language models available in multiple parameter sizes and includes both base and instruction-tuned variants [20]. The 3B instruction-tuned variant was selected for the present implementation. As an instruction-tuned model, it is designed to respond to natural-language instructions, which makes it suitable for the question-answering setting considered in this thesis.

Within the proposed RAG system, Qwen2.5-3B-Instruct receives both the user's question and retrieved context from the maritime technical manuals. The model uses this information to generate its response. This allows the generation process to be supported by information from the external document collection rather than relying exclusively on knowledge encoded within the model. The use of retrieved external information is particularly important because the internal knowledge of an LLM is subject to several limitations, which are discussed in the following section.

1.2 Limitations of Large Language Models

Despite their impressive capabilities, LLMs have several important limitations. Much of the knowledge available to an LLM is acquired during training and represented within the parameters of the model. This form of knowledge is commonly referred to as parametric knowledge [3]. Once training has been completed, updating the knowledge of the model generally requires additional training or model adaptation. Consequently, an LLM may not have access to information that was unavailable during its training or that has changed since then [22].

Another limitation concerns the accuracy and reliability of the information generated by LLMs. An LLM may generate a response that appears clear and convincing. However, the information in the response may be factually incorrect or unsupported. This problem is generally described as hallucination [21]. The possibility of such responses is particularly important to consider in applications where users depend on the model for factually accurate information [21].

These limitations can become particularly relevant when LLMs are applied to specialized technical domains. General-purpose language models may not contain the detailed or domain-specific information required to answer questions about particular technical systems, equipment, or documentation. In addition, their internal knowledge may be outdated when the required information has changed since training [22]. In such cases, relying exclusively on the model's parametric knowledge may be insufficient, especially when the required information is contained in documents that were not part of the model's training data.

General-purpose language models are not always directly suited to specialized applications. Different methods can therefore be used to adapt them to more specific tasks. These include prompt engineering, fine-tuning, and Retrieval-Augmented Generation [4]. Fine-tuning uses additional training to adapt a model to a particular task or domain. RAG follows a different approach by retrieving information from an external source and providing it to the model when generating a response. This approach is particularly useful when working with specialized knowledge or information that may need to be updated over time [4].

Among these approaches, RAG is particularly suitable for applications that depend on external or specialized knowledge [4]. Unlike fine-tuning, it does not require this external knowledge to be incorporated into the model through additional training. Instead, the knowledge can remain in an external source and be accessed when needed. This also makes it possible to update the external knowledge source without retraining the language model [3]. These characteristics make RAG particularly relevant to the present thesis, which focuses on the use of specialized maritime technical documentation. The RAG approach and its operation are discussed in the following section.

1.3 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) combines two types of knowledge. It uses the parametric knowledge of the language model together with information retrieved from an external source [3]. Instead of relying exclusively on the knowledge contained within the model's parameters, a RAG system retrieves information relevant to the user's query and provides it to the language model as additional context for response generation [3]. In this way, the generated response can incorporate both knowledge acquired during model training and information obtained from an external knowledge source.

The RAG framework consists of two main components: a retriever and a generator. The retriever searches an external knowledge source and identifies information that is relevant to the user's query. The retrieved information is then provided to the generator, which uses it as supplementary context when producing the final response [3]. The knowledge encoded within the parameters of a language model is referred to as parametric knowledge, whereas information obtained from an external source can be provided through non-parametric memory [3].

In practical RAG systems, external documents are commonly divided into smaller textual units, or chunks, to support the retrieval of relevant information. The way in which documents are divided can influence retrieval performance, since the retrieved units determine what information is subsequently made available to the language model [4], [23]. During retrieval, the user's query is compared with the indexed document content in order to identify relevant chunks, which are then provided to the language model as additional context for response generation [23].

This combination allows a language model to access information beyond the knowledge stored within its parameters. The additional information can be used without further training of the model. In the original RAG framework, Lewis et al. demonstrated that combining parametric and non-parametric memory can improve performance on knowledge-intensive natural-language processing tasks and can produce more factual language than parametric-only approaches [3]. The effectiveness of a RAG system, however, also depends on whether the retrieval component is able to identify information that is relevant to the user's query.

In the context of this thesis, RAG provides a suitable framework for allowing a language model to access information contained in maritime technical manuals. Rather than expecting the language model to contain detailed knowledge of specific maritime equipment within its parameters, relevant information can first be retrieved from the available manuals and then supplied to the model for response generation. This approach forms the basis of the system developed in this study and is described in detail in Chapter 2.

1.4 Challenges of Technical Information Retrieval in the Maritime Industry

Shipboard operating and maintenance manuals contain written procedures together with technical information regarding onboard equipment and systems. These manuals provide reference information and background knowledge that support seafarers in understanding and performing shipboard procedures [9].

Finding the required information within such documentation can be challenging. Seafarers often search manuals for specific information rather than reading them sequentially, while the structure and organization of technical documentation can further affect the process of locating and interpreting the required information [9]. The quality and accessibility of shipboard procedures are particularly important because deficiencies in operating and maintenance documentation have been identified as contributing factors in maritime incidents [9].

RAG is well-suited for applications that require access to specialized technical knowledge, as relevant information can be retrieved from technical documents before a response is generated [6], [7]. Recent studies have investigated the use of RAG for question answering over specialized technical documentation [6]-[8]. This includes research in the maritime domain, where RAG has been applied to retrieve and process information from engineering documentation [8]. These studies are reviewed in the following section in relation to the development of the proposed maritime RAG system.

One representative study was conducted by Soman and Roychowdhury [6], who investigated the challenges of developing Retrieval-Augmented Generation systems for technical documents. Rather than proposing a new RAG architecture, the authors experimentally evaluated how different retrieval design choices affect question-answering performance over specialized technical documentation.

The authors conducted their experiments using IEEE wireless networking standards and a technical glossary containing battery terminology. The evaluation was based on 42 representative domain-specific questions and examined factors including chunk length, keyword-based search, embedding and retrieval strategies, and the ranking of retrieved results within the RAG pipeline.

The study showed that technical documentation presents specific challenges for retrieval in RAG systems, particularly because embeddings may not adequately capture domain-specific information. The authors found that larger text chunks reduce the reliability of sentence embeddings, while retrieval performance depends not only on semantic similarity but also on appropriate document segmentation and retrieval strategies. In addition, sentence-level similarity combined with paragraph-level retrieval was found to provide richer context for response generation and improve the quality of generated answers.

This work is particularly relevant to the present thesis because both studies focus on question answering over technical documentation. However, while Soman and Roychowdhury primarily investigate retrieval strategies through experimental evaluation, the proposed system implements a complete locally deployed RAG pipeline for maritime technical manuals. Their findings on document segmentation and retrieval quality also provide a useful comparison with the retrieval approach used in the proposed maritime RAG system.

Pu et al. [7] proposed a customized Retrieval-Augmented Generation (RAG) framework for question answering over Electronic Design Automation (EDA) tool documentation. The study focuses on the difficulties of applying general-purpose RAG systems to this type of specialized technical documentation.

The proposed system introduces domain-specific improvements at different stages of the RAG pipeline. The text embedding model is fine-tuned using contrastive learning. The system also uses a reranker distilled from a proprietary LLM. Finally, the generative language model is fine-tuned using domain-specific data. To evaluate the proposed approach, the authors also introduced the ORD-QA benchmark. This is a domain-specific dataset designed for question answering over OpenROAD documentation.

The experimental results showed that the customized RAG system performed better than the evaluated baseline approaches. Improved performance was reported on both the ORD-QA benchmark and a commercial EDA tool. The results demonstrate the potential benefits of adapting different components of a RAG system to a specialized domain.

Although Pu et al. [7] focus on a different technical domain, their work also applies RAG to question answering over specialized technical documentation. While the proposed framework targets EDA tool documentation, the present work focuses on maritime technical manuals. The study demonstrates the potential advantages of adapting different components of a RAG pipeline to specialized technical domains and provides a useful point of comparison for the present work.

Galvan et al. [8] proposed a multi-agent framework combining Large Language Models and Retrieval-Augmented Generation for processing and analyzing shipbuilding engineering documents. The authors focused on the difficulty of retrieving information from this type of complex technical documentation. Relevant content can appear in different sections of a document and may take several forms, including text, tables, equations, and diagrams.

The proposed system was evaluated using technical documentation from the marine division of Bureau Veritas, specifically a chapter containing requirements and specifications related to ship structures. The document includes technical information in multiple formats, such as tables,

engineering equations, and diagrams. Prior to retrieval, the technical documentation was processed using an optical character recognition (OCR) system and converted into Markdown format.

To address some of the limitations of conventional RAG systems, the authors proposed an architecture based on five agents: the Chapter-Suggester, Main Retriever, Prompt-Generator, Organizer, and Response-Giver. Rather than relying on a single retrieval step, the system first uses the Chapter-Suggester to identify sections of the manual that may contain relevant information. The Main Retriever then searches the embedded document chunks stored in a vector database. The other agents support the subsequent processing of the retrieved material. They identify information that may still be missing, combine content from different sections, and prepare the final response. This approach is particularly useful when the information needed to answer a question is located in different parts of a technical document.

Experimental evaluation examined both information retrieval and engineering equation-related tasks. The results demonstrated that the system was capable of retrieving and interpreting domain-specific information, although performance varied across the evaluated language models and task complexity. In particular, complex engineering calculations remained challenging. The reported difficulties included multi-step reasoning errors, incomplete retrieval of relationships between document sections, and limitations in numerical reasoning. The study therefore highlights both the capabilities and the current limitations of LLM-based RAG systems for engineering applications.

A key similarity with the present work is the use of Retrieval-Augmented Generation for maritime technical documentation. While Galvan et al. focus on shipbuilding engineering standards and employ a multi-agent architecture capable of integrating information from multiple document sections, the present work develops a lighter locally deployed RAG system for question answering over maritime technical manuals. The study demonstrates the applicability of RAG to maritime engineering documentation, while also highlighting the importance of reliable retrieval and appropriate handling of complex technical content.

Nam et al. [10] proposed a multimodal Retrieval-Augmented Generation system for question answering over technical maintenance manuals.

The study focuses on automotive maintenance documentation and investigates how domain-specific retrieval and model adaptation can improve the generation of answers based on specialized technical information.

The proposed system was developed using a 522-page Hyundai Staria maintenance manual containing both textual and visual information. The manual was processed and segmented into smaller units, which were subsequently used to construct a domain-specific question-answering dataset. The authors generated 5,065 question-answer pairs and employed embedding-based retrieval to identify relevant segments of the technical documentation for each user query.

In addition to the retrieval component, the framework incorporates multimodal information and applies LoRA-based fine-tuning to adapt the language model to the maintenance domain. To improve retrieval in the specialized maintenance domain, the proposed architecture incorporates sentence-level similarity annotations together with the BAAI-bge-m3 embedding model.

Experimental evaluation showed that the proposed approach improved answer quality compared with the evaluated baseline configurations, with improvements reported across cosine similarity, BERTScore, and ROUGE-L. The results indicate that combining retrieval with domain-specific adaptation can improve question answering over specialized maintenance documentation.

The use of RAG for question answering over specialized technical manuals provides a clear connection with the present work. However, Nam et al. focus on automotive maintenance documentation and employ a multimodal architecture together with LoRA-based model adaptation, whereas the present work focuses on maritime technical manuals and implements a lighter locally deployed RAG system. The study therefore provides further evidence of the applicability of RAG to technical maintenance documentation while illustrating a more complex approach based on multimodal processing and model adaptation.

Rasaq et al. [11] examined the use of an edge-deployed RAG system for accessing technical maintenance information in aerospace asset health management. Their work considers situations in which cloud connectivity is limited or unavailable, with the RAG pipeline running locally on resource-constrained edge hardware.

The proposed architecture processes maintenance documents through a pipeline consisting of document ingestion, text segmentation, embedding generation, vector indexing, semantic retrieval, reranking, and response generation. The system uses FAISS for vector-based retrieval and a locally hosted Llama 3.2 3B language model for answer generation, allowing the entire retrieval and generation process to operate offline.

The evaluation was based on aerospace maintenance documentation and FAA Airworthiness Directives, which were processed and indexed locally to form the knowledge base used by the RAG pipeline. Evaluation on the Jetson Orin Nano resulted in an average fidelity score of 0.83, with inference times generally ranging from approximately 4 to 10 seconds. These results demonstrated that the complete RAG pipeline could operate on edge hardware and generate answers based on retrieved maintenance information without relying on cloud-based services.

Local deployment is an important point of connection between this study and the present work. While Rasaq et al. focus on aerospace maintenance documentation and deployment on resource-constrained edge hardware, the present work focuses on question answering over maritime technical manuals. The study demonstrates the feasibility of performing document retrieval and response generation entirely offline and provides a useful point of comparison for the local deployment approach adopted in the proposed maritime RAG system.

Wulf et al. [12] presented the design and evaluation of a Retrieval-Augmented Generation system for guided operations and maintenance in an industrial environment. The system was developed in collaboration with a production machine manufacturer and aims to provide machine operators with on-demand access to technical knowledge contained in the manufacturer's documentation. Users can submit operational or maintenance-related problems in natural language. The system then retrieves relevant information from the available documentation and uses it to generate context-based responses.

The knowledge base included the full technical documentation for two machine models, with manuals available in both English and German. All processed documents were stored locally in ChromaDB, and the prototype did not depend on external APIs.

Three retrieval methods were tested: semantic vector search, BM25 keyword search, and a hybrid method combining both. The hybrid method produced the best retrieval results and was therefore used in the final RAG pipeline.

The system was evaluated using 400 real-world customer questions together with relevant sections of the technical documentation and expert-verified ground-truth answers. The evaluation considered conformance with the provided context, answer completeness, and response latency. The prototype achieved over 90% conformance with the source documentation and more than 80% answer completeness. A separate manual evaluation by technical service experts found that 81% of the reviewed responses were entirely correct and contextually appropriate for practical use.

The study also identified limitations related to document processing and retrieval. Page-based chunking occasionally separated logically connected information, resulting in incomplete retrieved context, while irrelevant retrieved information could contribute to unsupported responses. In addition, the text-only architecture could not process technical information contained in diagrams and images.

The focus on question answering over specialized technical documentation and local deployment closely relates this study to the present work. While Wulf et al. focus on guided operations and maintenance for industrial production machinery, the present work applies RAG to maritime technical manuals. Their findings regarding retrieval strategies, document segmentation, and local deployment provide useful points of comparison for the design of the proposed maritime RAG system.

Han et al. [13] developed a RAG-based question-answering system for ship regulations. Their work focused on the Korean Register Rules, a large collection of regulatory documents where finding the appropriate information can be difficult because of the volume, structure, and specialized maritime terminology involved.

The material used for the system consisted of 68 PDF documents, covering approximately 10,400 pages and 10,100 articles. The regulations were divided according to their hierarchical structure instead of being treated simply as fixed-length text segments. Semantic retrieval was then used to match user questions with relevant regulatory content. The retrieval model was also adapted to the maritime domain so that terminology specific to ship regulations could be represented more effectively.

Evaluation was carried out with 60 queries prepared in collaboration with the Korean Register and industry professionals, along with expert answers and the relevant rules for each query. After domain adaptation, the retrieval model obtained an accuracy score of 94.33 out of 100. LLaMA 3.1 was used to generate the answers, while a quantized configuration was also examined to reduce computational requirements. The system could therefore work with locally stored regulatory information without requiring an online connection.

The maritime-specific focus of this study makes it closely related to the present work. However, Han et al. focus on ship regulations and employ domain-specific retrieval adaptation and document-structure-aware processing, whereas the present thesis focuses on maritime technical manuals and implements a lighter RAG pipeline without domain-specific model adaptation. The study therefore provides a useful maritime-specific point of comparison and demonstrates the potential benefits of adapting retrieval to specialized maritime terminology.

Ren et al. [14] proposed MARTIAN (Maritime Agent for Real-time Tactical Inference And Navigation), a maritime decision-support system designed for deployment on Maritime Autonomous Surface Ships (MASS). The system is based on Qwen2.5-14B-Instruct and combines supervised fine-tuning, Retrieval-Augmented Generation, and function calling. Supervised fine-tuning is used to develop maritime procedural reasoning, RAG provides access to external maritime knowledge, and function calling enables the system to interact with vessel data through deterministic engineering tools. The overall architecture was designed for local edge deployment on vessel hardware.

The RAG component searches a knowledge base represented through vector embeddings, covering 15 maritime domains. BGE-M3 is used locally to encode the document chunks, which are matched with the user query based on cosine similarity. For each query, the three highest-ranked chunks are passed to the language model as supplementary context. The stored material mainly covers regulatory and factual maritime knowledge, while procedural knowledge is addressed through supervised fine-tuning.

Evaluation was based on the BM-MCQ benchmark, which contains 750 questions from four maritime domains. MARTIAN reached an overall accuracy of 73.23% with supervised fine-tuning alone, increasing to 81.16% when retrieval was added.

The improvement was not uniform across all question categories. Maritime safety and navigational systems benefited from the additional retrieved information, whereas a slight decrease was reported for logic-intensive core navigation tasks. The authors also tested different amounts of retrieved context. They found that using more chunks could introduce semantic interference and negatively affect reasoning. This suggests that providing more retrieved information is not necessarily beneficial when some of that information is less relevant to the question.

The combination of a maritime application and local edge deployment makes this work closely related to the present thesis. MARTIAN was deployed on an NVIDIA Jetson AGX Orin using INT4 quantization, demonstrating the feasibility of operating a RAG-enhanced maritime language model on local hardware without relying on a cloud-based language model service. Although the system focuses on maritime decision support and regulatory knowledge rather than question answering over technical manuals, it provides a useful example of a locally deployed RAG architecture in maritime applications.

Overall, the reviewed studies demonstrate the potential of RAG for improving access to information contained in specialized technical documentation. Existing approaches have investigated different aspects of the RAG pipeline, including document segmentation and retrieval strategies [6], domain-specific adaptation [7], [13], multimodal processing [10], and local or edge deployment [11]–[14]. Within the maritime domain, RAG has been applied to shipbuilding engineering documentation [8], ship regulations [13], and maritime decision support for autonomous surface ships [14]. These studies demonstrate the applicability of RAG to different types of specialized maritime information.

The reviewed systems also use different levels of architectural complexity. These include multi-agent frameworks [8], domain-adapted retrieval [13], multimodal and fine-tuned approaches [10], and systems combining supervised fine-tuning with function calling [14]. This provides the motivation for investigating a simpler and lightweight RAG architecture in the present thesis. The proposed system focuses on locally deployed question answering over maritime technical manuals, without requiring domain-specific model adaptation or more complex multi-agent and multimodal architectures.

Study	Application Domain	Main Approach	Retrieval Approach	Deployment Approach	Main Relevance to Present Thesis
Soman & Roychowdhury [6]	Technical documentation	Experimental analysis of RAG design choices	Embedding-based and keyword-based retrieval; evaluation of chunk size and sentence-/paragraph-level retrieval	Not specified	Demonstrates how chunking and retrieval strategy affect RAG performance on specialized technical documents
Pu et al. [7]	EDA tool documentation	Domain-customized RAG with adapted embedding, reranker, and language models	Domain-adapted embedding retrieval with fine-tuned reranking	Not specified	Demonstrates the benefits of domain-specific adaptation and reranking for QA over specialized technical documentation
Galvan et al. [8]	Shipbuilding engineering documentation	Multi-agent RAG framework	Vector-based retrieval coordinated through a multi-agent pipeline	Not specified	Demonstrates the applicability of RAG to maritime engineering documentation and retrieval across multiple document sections
Nam et al. [10]	Automotive maintenance manuals	Multimodal RAG with LoRA-based domain adaptation	Semantic embedding-based retrieval with sentence-level similarity and linked visual content	Not specified	Demonstrates RAG for technical maintenance manuals and shows the benefits of multimodal information and domain adaptation
Rasaq et al. [11]	Aerospace maintenance documentation	Edge-deployed RAG framework	Dense vector retrieval with cross-encoder reranking	Fully offline / edge-deployed	Demonstrates the feasibility of running a complete RAG pipeline locally on resource-constrained hardware
Wulf et al. [12]	Industrial operations and maintenance	RAG for guided operations and maintenance	Semantic, BM25, and hybrid ensemble retrieval	Locally Deployed	Demonstrates locally deployed RAG for QA over technical documentation and compares alternative retrieval strategies
Han et al. [13]	Ship regulations	Domain-adapted maritime RAG	Semantic similarity retrieval with a domain-adapted bi-encoder	Fully offline / local	Demonstrates maritime-specific local RAG and the benefits of adapting retrieval to specialized maritime terminology
Ren et al. [14]	Maritime autonomous surface ships	Edge maritime LLM agent combining SFT, RAG and function calling	Hybrid keyword and dense retrieval using BGE-M3 embeddings; Top-3 chunks	Edge/local (NVIDIA Jetson AGX Orin)	Demonstrates the feasibility of RAG and local edge deployment in a maritime application
Present Thesis	Maritime technical manuals	Lightweight RAG for technical manual QA	Semantic vector retrieval with sentence-level snippet selection	Fully local / offline	Provides a lightweight locally deployed RAG approach for QA over maritime technical manuals with source traceability

Table 1.1: RAG Systems for specialized technical documentation

The comparison presented in Table 1.1 shows that existing RAG systems for specialized technical domains follow considerably different approaches depending on the requirements of their target application. Several studies introduce additional architectural components, such as domain-specific model adaptation [7], [13], multimodal processing [10], multi-agent retrieval [8], hybrid retrieval strategies [12], or the combination of supervised fine-tuning, hybrid retrieval, and function calling [14]. Other studies emphasize local, offline, or edge operation, demonstrating the feasibility of deploying RAG-based systems without continuous dependence on cloud-based language model services [11]–[14].

Previous maritime applications of RAG have focused on areas such as shipbuilding engineering documentation [8], ship regulations [13], and decision support for autonomous surface ships [14]. These applications differ from the technical-manual question-answering task considered here, both in the type of information involved and in their system requirements. Some also rely on domain adaptation or more elaborate retrieval and processing methods. In this thesis, a more lightweight setup is considered, with the focus placed specifically on retrieving information from maritime technical manuals. The implemented RAG system runs locally and uses an existing language model without domain-specific fine-tuning or access to external language-model APIs.

1.5 Aim and Contribution of the Thesis

The aim of this thesis is to design, implement, and evaluate a locally deployed Retrieval-Augmented Generation system for question answering over maritime technical manuals. The proposed system is intended to provide users with direct access to relevant technical information through natural-language questions, offering an alternative to manually searching through extensive technical documentation.

To achieve this objective, maritime technical manuals are processed and divided into overlapping text segments, which are converted into vector representations using the all-MiniLM-L6-v2 SentenceTransformer model. The resulting embeddings are stored in a ChromaDB vector database together with document and page metadata. When a user submits a question, the same embedding model is used to represent the query, and semantic retrieval is performed to identify relevant information from the available manuals.

The retrieved information is subsequently used to construct the context provided to the Qwen2.5-3B-Instruct language model. The model is instructed to rely on the retrieved manual content when generating its response. It is also instructed to indicate when sufficient information is not available. Both information retrieval and response generation are performed locally, without relying on external language model APIs. The system also preserves source information through manual and page metadata. This allows generated responses to be associated with the technical documentation from which the supporting information was retrieved.

The main contribution of this thesis is the development and evaluation of a lightweight RAG pipeline for maritime technical manuals that runs locally. Unlike some of the more complex approaches found in the literature, the system does not use domain-specific model fine-tuning, multi-agent architectures, or multimodal processing. Instead, it combines semantic retrieval with local language-model generation. The aim is to examine how effectively a relatively simple RAG setup can provide access to specialized maritime technical information.

The evaluation considers the two main stages of the pipeline separately. First, it examines whether the retrieved context contains the information needed to answer each question. Second, it examines whether this information is correctly reflected in the generated answer. Four technical manuals with different content and sizes are used for the evaluation. This provides a basis for examining the performance and limitations of the lightweight approach across different types of maritime technical documentation.

Chapter 2 – System Design and Implementation

2.1 Overall System Architecture

The system developed in this thesis follows a Retrieval-Augmented Generation (RAG) architecture designed for question answering over maritime technical manuals. The system combines document processing, semantic information retrieval, and local language-model generation in order to provide answers grounded in information extracted from the available technical documentation. Rather than relying on the language model alone, relevant information is retrieved from the indexed manuals for each user question and provided as context for response generation.

The overall architecture can be divided into two main stages: document indexing and question answering. The document indexing stage is performed before user queries are processed and is responsible for transforming the available PDF manuals into a searchable knowledge base. During this stage, text is extracted from each page of the manuals, cleaned, divided into overlapping text chunks, and converted into vector embeddings using the all-MiniLM-L6-v2 SentenceTransformer model. The resulting embeddings are stored in a persistent ChromaDB vector database together with the corresponding text chunks and metadata, including the source document, page number, document title, and chunk index.

The question-answering stage begins when the user submits a natural-language question. The question is converted into a vector embedding using the same SentenceTransformer model and submitted to the ChromaDB collection, where semantic retrieval is performed against the stored document embeddings. The system retrieves the most relevant document chunks together with their metadata and similarity distances.

The retrieved chunks are subsequently processed to construct the context provided to the language model. For each retrieved chunk, the system identifies the sentence most semantically related to the user's question and extracts a surrounding text snippet. These snippets are combined with the corresponding document title and page number and assembled into the context used to construct the final prompt. The prompt also contains explicit instructions requiring the language model to base its response only on the retrieved manual information and to return a predefined response when sufficient information is not available.

Response generation is performed using the Qwen2.5-3B-Instruct language model through the Hugging Face Transformers text-generation pipeline. After generation, the response is processed before being presented to the user. When the query-processing script is executed directly through the Python environment, the output also includes information associated with the retrieved sources, such as the manual title, page number, similarity distance, and relevant text snippet. This detailed output allows the retrieved content to be inspected and related to the corresponding technical documentation.

A key design characteristic of the proposed architecture is that both information retrieval and language-model inference are performed locally. The ChromaDB database is stored persistently in the local execution environment, while Qwen2.5-3B-Instruct is executed locally through the Hugging Face Transformers text-generation pipeline. Consequently, the question-answering process does not rely on an external language-model API. The individual components and processing stages of the proposed architecture are described in greater detail in the following sections.

2.2 Document Processing and Indexing

The first stage of the system focuses on processing the technical manuals before they can be used for information retrieval. During this stage, the documents are prepared for indexing through a series of processing steps. These steps include text extraction, text preprocessing, text chunking, embedding generation, and storage in a vector database.

The system processes the technical manuals provided in PDF format and extracts the text from each page individually using the *pypdf* library. Pages from which no text can be extracted are skipped. For each processed page, the system retains the source document path, page number, document title, and extracted text. Before further processing, repeated whitespace and line breaks in the extracted text are replaced with single spaces.

After preprocessing, the text extracted from each page is divided into smaller sections, referred to as chunks. Splitting the documents into chunks allows the system to retrieve relevant text segments instead of processing entire manuals for every query [4]. In the implemented system, each chunk contains up to 1,200 words, while consecutive chunks overlap by 150 words. These values were selected as a practical configuration intended to retain sufficient surrounding technical context while still dividing the manuals into retrievable sections of manageable size. The overlap further reduces the risk of separating related information when relevant content occurs near a chunk boundary. Chunks containing ten words or fewer are discarded because they provide very limited textual context for subsequent retrieval.

Each text chunk is converted into a vector embedding using the SentenceTransformer model *all-MiniLM-L6-v2*. The selected model is part of the Sentence-Transformers framework, which builds on the Sentence-BERT (SBERT) approach proposed by Reimers and Gurevych [5]. Sentence-BERT is designed to generate semantically meaningful text representations, making it particularly suitable for semantic search and information retrieval tasks [5]. The model was selected as a lightweight embedding solution suitable for the locally deployed architecture of the proposed system. Its use enables semantic representation of both document chunks and user queries without requiring a separate domain-specific embedding model, while keeping the retrieval pipeline comparatively simple. In the implemented indexing process, embeddings are generated in batches of up to 128 chunks before being added to the vector database.

In this work, the vector database is implemented using a persistent ChromaDB collection. The collection is configured to use cosine distance for similarity comparison. A persistent local vector store was selected so that the indexed manuals could remain available across executions without requiring repeated embedding generation. Cosine distance was used as the similarity measure for comparing the semantic representations of queries and document chunks within the retrieval process.

ChromaDB stores each chunk together with its embedding and metadata, including the source document path, page number, document title, and chunk index. A unique identifier is created from the document name, page number, and chunk number, making it possible to trace the indexed content back to its location in the original technical manual.

Before a new indexing process begins, any existing ChromaDB collection with the same name is removed and a new collection is created. The available PDF manuals are then processed and indexed in the new collection. Once the indexing process has been completed, the resulting persistent vector collection forms the searchable knowledge base used during query processing and information retrieval, as described in the following section.

2.3 Query Processing and Retrieval

After the document collection has been indexed, the system is able to process user queries and retrieve relevant information from the technical manuals. When a user submits a natural-language question, the question is converted into a vector embedding using the same all-MiniLM-L6-v2 SentenceTransformer model that was used during the indexing stage. Using the same embedding model for both document chunks and user queries ensures that they are represented within the same semantic vector space, allowing their representations to be compared during semantic retrieval [5].

The generated query embedding is submitted to the persistent ChromaDB collection, where semantic similarity search is performed against the stored document embeddings [2]. The collection uses cosine distance for vector comparison. For each query, the system retrieves the six most relevant document chunks together with their metadata and similarity distances. A retrieval depth of six was used to provide multiple candidate passages while keeping the amount of retrieved information limited. The retrieved chunks are then processed to identify the most relevant text snippets for context construction. Metadata such as the document title and page number is preserved so that the retrieved information can be traced back to its source.

After the relevant document chunks have been retrieved, the system constructs the context that will be provided to the language model. Each retrieved chunk is divided into sentences. The user's question and the candidate sentences are represented using the same SentenceTransformer model. The sentence with the highest semantic similarity to the question is then identified. Instead of using the entire retrieved chunk, the system extracts a shorter text snippet around the selected sentence. The snippet includes up to approximately 400 characters before and after the selected sentence. This preserves useful surrounding information while limiting the amount of unrelated text included in the context. The resulting snippets are then used for context and prompt construction.

Each selected text segment is then combined with its document title and page number, allowing the retrieved information to remain connected to its original source. The selected snippets are finally joined into a single context, which is used to construct the prompt provided to the language model. Following the RAG approach, the retrieved context and the user's question are combined to form the prompt [2].

The prompt contains specific instructions that limit the answer to the supplied information and request a concise response without unnecessary repetition. If the manual content does not contain enough information to answer the question, the model is instructed to return a predefined message.

2.4 Response Generation

The completed prompt is passed to Qwen2.5-3B-Instruct for response generation. The model uses the retrieved manual content to formulate an answer to the user's question. This helps ground the generated response in information retrieved from the technical documentation rather than relying primarily on knowledge already encoded in the model.

Qwen2.5-3B-Instruct was selected as a comparatively compact instruction-tuned model for the locally deployed architecture. The model can be executed on the available local hardware and provides the language-generation component of the RAG pipeline.

Although Qwen2.5-3B-Instruct possesses knowledge acquired during its original pre-training, it was not specifically trained or fine-tuned on the maritime technical manuals used in this thesis. The retrieved document context and the instructions included in the prompt are therefore used to guide response generation towards the information contained in the indexed documentation.

The language model is executed locally through the Hugging Face Transformers text-generation pipeline. During response generation, the `max_new_tokens` parameter is set to 200. This limits the maximum number of new tokens that can be generated for each response. In addition, stochastic sampling is disabled (`do_sample=False`). This favors consistent and reproducible responses during system use and experimental evaluation, rather than introducing variation through random sampling. The model is configured to run on the CPU. This allows response generation to be performed locally without relying on an external language-model API.

Before the generated answer is presented to the user, it is further processed to produce a concise final response. The generated text is first cleaned by removing unnecessary formatting and redundant whitespace and is then divided into individual sentences. Very short text fragments are excluded from the final response. The implementation retains up to the first three valid sentences of the generated response. The model is instructed to return a predefined fallback message when the provided context does not contain sufficient information to answer the question. If this fallback message appears together with an otherwise valid generated response, it is removed during post-processing in order to avoid presenting contradictory information to the user.

When the query-processing script is executed directly, the generated answer is presented together with detailed information from the retrieved sources, including the corresponding technical manual title, page number, similarity distance, and relevant text snippet. This output provides a more detailed view of the retrieval results and allows them to be compared with the original technical documentation.

2.5 Overall System Workflow

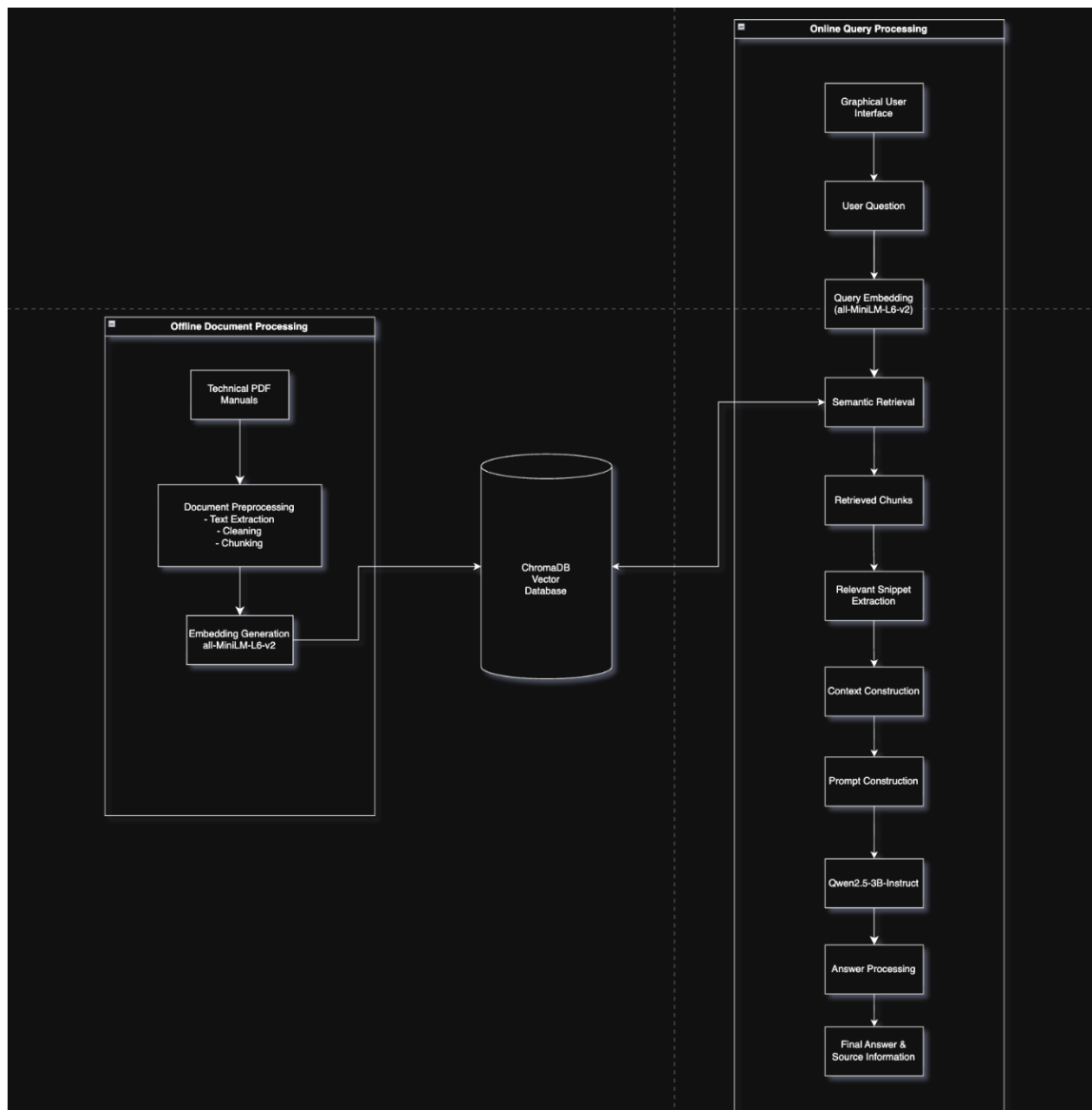


Figure 2.1 Overall workflow of the proposed RAG system

Figure 2.1 provides an overview of the complete workflow of the proposed system, bringing together the individual processing stages described in Sections 2.2–2.4. As illustrated, the architecture separates the preparation of the technical documentation from the processing of user queries. The document-processing stage creates the persistent vector-based knowledge base, while the query-processing stage uses this indexed information to retrieve relevant content and generate a response grounded in the technical documentation.

Once the document collection has been indexed, a user question follows a sequential path through query embedding, semantic retrieval, relevant snippet extraction, context and prompt construction, and response generation. The generated output is subsequently processed before the final answer is presented to the user. The final answer is presented together with the corresponding source information, allowing the retrieved evidence to be traced back to the original technical documentation.

2.6 Local Deployment and Execution Environment

The developed system is designed to operate entirely within a local computing environment, without relying on an external language-model API for response generation. Local deployment was treated as a design requirement of the proposed architecture rather than only as an implementation constraint. Both the retrieval and generation stages are therefore performed locally. This allows the technical documentation and retrieved information to remain within the local system throughout the processing workflow. It also reduces dependence on continuous access to external cloud-based language-model services.

The system was implemented in Python 3.11.14 using Spyder and tested on a 16-inch MacBook Pro (2021) equipped with an Apple M1 Pro processor and 16 GB of unified memory. All experiments were performed on macOS, with Qwen2.5-3B-Instruct running on the CPU. The available hardware was sufficient to run the complete RAG pipeline locally, without the use of a dedicated GPU or an external language-model service.

The implementation uses Sentence-Transformers to generate vector representations with the all-MiniLM-L6-v2 model, while ChromaDB provides persistent local storage for the document chunks, embeddings, and associated metadata. The Qwen2.5-3B-Instruct language model is loaded and executed locally through the Hugging Face Transformers framework. As a result, the main stages of the pipeline, including semantic retrieval, context construction, and response generation, are performed within the local execution environment.

The ChromaDB collection remains stored on the local file system after the manuals have been indexed. It can therefore be used for later queries without processing the documents and generating their embeddings again. At query time, the existing collection is accessed to retrieve the document chunks needed for response generation.

A graphical user interface was additionally implemented to provide a more accessible means of interacting with the developed RAG system. The interface operates as a user-facing layer over the same locally executed retrieval and response-generation pipeline described in the preceding sections. Users can submit natural-language technical questions and receive generated responses together with information identifying the corresponding retrieved sources. The design and operation of the graphical interface are described in Section 2.7.

2.7 Graphical User Interface

A graphical user interface was developed to provide a direct way for users to interact with the implemented maritime RAG system. Until this stage, the individual components of the system were executed and tested through the Python environment. The addition of the graphical interface provides a simpler way to use the complete system, without requiring questions to be entered directly through the Python console. It was developed in Python using CustomTkinter and follows a conversation-style layout, where submitted questions and generated answers remain visible within the same session.

Figure 2.2 shows the graphical interface of the developed system. The main area is used for the interaction between the user and the system. A text field at the bottom of the window allows the user to enter a technical question, which is submitted through the Ask button. The left-hand side contains the system identification and a New Chat option. When New Chat is selected, the current conversation is cleared and a new interaction can begin.

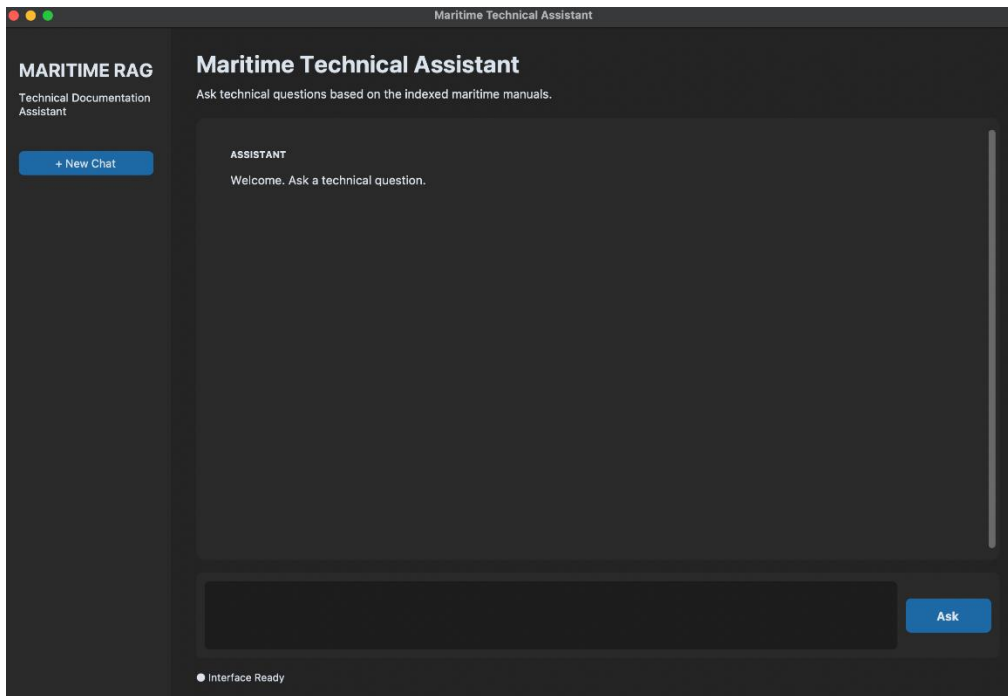


Figure 2.2 Graphical user interface of the developed maritime RAG system

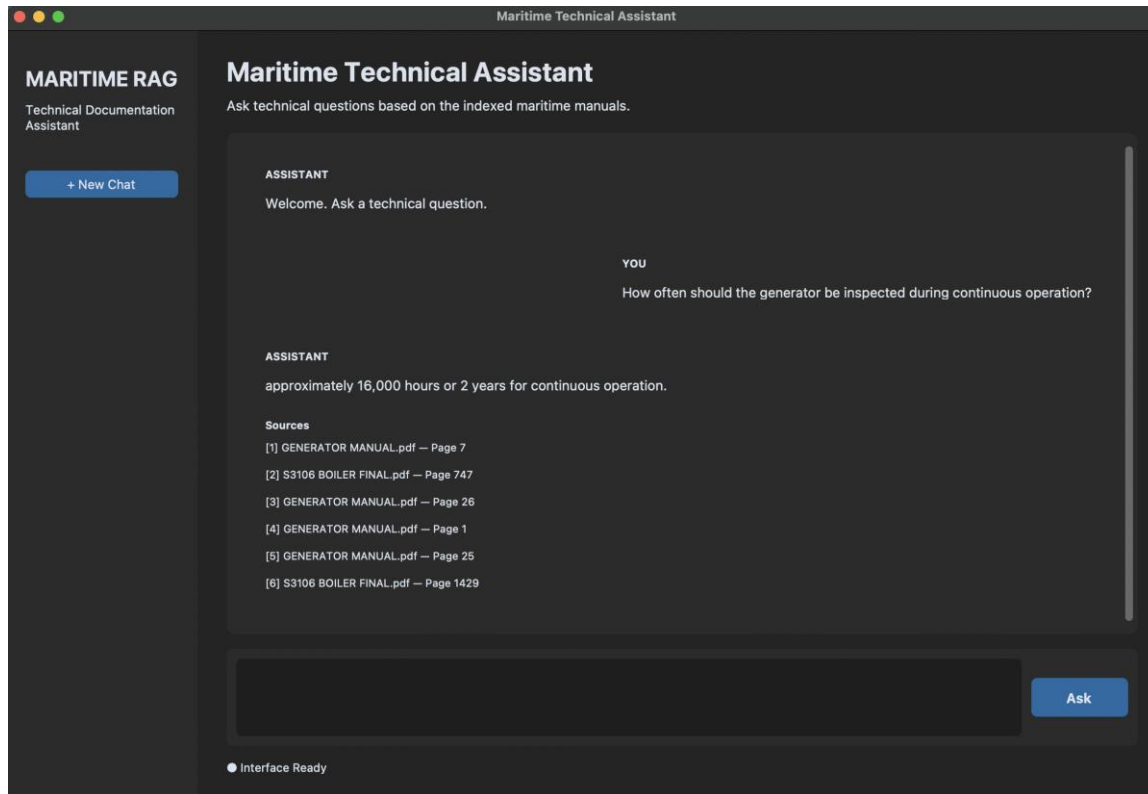


Figure 2.3 Example query and response in the UI

The graphical interface is connected directly to the RAG pipeline described in the previous sections. When the user submits a question, the same query-processing procedure described in Section 2.3 is used. The question is converted into an embedding and used to retrieve the six highest-ranked document chunks from the existing ChromaDB collection. The retrieved chunks then follow the same processing sequence described earlier, including relevant text extraction, context construction, and prompt creation. The resulting prompt is passed to the locally executed Qwen2.5-3B-Instruct model, and the generated output is processed before being presented to the user.

The generated answer is shown in the conversation area together with information about the retrieved sources. For each result, the interface shows its ranked position, the title of the corresponding technical manual, and the page number. The similarity distance used internally during retrieval is not included in the user-facing display, since it is more useful during technical analysis and evaluation. The manual title and page number allow the user to locate the relevant information in the original documentation and compare it with the generated response when necessary.

During question processing, the interface also indicates the current state of the system. For example, it shows when the manuals are being searched and when a response is being generated. Since local language-model inference may require some processing time, question processing is performed in a separate thread. This keeps the graphical interface responsive while retrieval and response generation are taking place. Once processing is complete, the generated answer and the corresponding source information are added to the conversation, and the system is ready to receive the next question.

Chapter 3 – Experimental Evaluation

3.1 Evaluation Methodology

The developed Retrieval-Augmented Generation system was evaluated through a series of controlled question-answering tests using maritime technical documentation. The purpose of the evaluation was to examine the performance of the complete RAG pipeline, from the retrieval of relevant information to the generation of the final response. Since the effectiveness of a RAG system depends on both stages, retrieval performance and answer quality were considered separately during the evaluation [15].

The evaluation was conducted using four maritime technical manuals covering different shipboard systems and technical domains. All documents were processed and indexed using the document-processing pipeline described in Chapter 2. A set of 24 evaluation questions was then used, with six questions corresponding to each manual. The questions were based on information contained in the documentation and were designed to represent different types of technical information that may be requested when consulting maritime manuals.

For each evaluation question, the expected information was identified from the corresponding technical manual before the system response was assessed. The question was then submitted to the system in natural language using the same retrieval and response-generation process for all tests. For each query, the system retrieved the six passages considered most relevant according to semantic similarity and generated a final answer based on the retrieved context. The retrieved passages, their ranked positions, and the final generated answer were then recorded for evaluation.

Two aspects were examined for every evaluation question. First, the retrieved passages were manually inspected to determine whether they contained the information required to answer the question. When the relevant information was found, its position within the ranked retrieval results was recorded. The generated answer was then manually compared with the reference information identified in the corresponding technical manual. This made it possible to distinguish between cases in which the required information was not successfully retrieved and cases in which the information was available in the retrieved context but was not correctly reflected in the generated response.

Sections 3.2 and 3.3 describe the technical manual dataset and the evaluation questions. Section 3.4 then defines the criteria used to assess retrieval and answer quality, before the results and their analysis are presented in the following sections.

3.2 Technical Manual Dataset

The evaluation dataset consisted of four maritime technical manuals representing different types of shipboard equipment and technical documentation. The dataset was intentionally composed of manuals covering different operational and engineering domains in order to evaluate the system using heterogeneous technical content rather than restricting the experiments to a single type of equipment.

The selected documentation consists of a synchronous generator operating manual, an ECDIS operation manual, a maintenance manual for an MFC valve, and a boiler technical manual. These documents cover different areas of maritime operations and engineering, including electrical equipment, navigation systems, mechanical components, and boiler systems. This diversity provides an appropriate basis for examining the ability of the proposed RAG system to retrieve and interpret technical information across different types of maritime documentation.

Technical Manual	Technical Domain	Pages	Indexed chunks
Generator Manual	Synchronous Generator/Electrical	56	56
ECDIS Instruction Manual-Operation	Navigation/ECDIS	999	978
Maintenance Manual (MFC Valve)	Mechanical/Valve Maintenance	6	5
S3106 Boiler Manual	Boiler/Marine Engineering	1787	1774

Table 3.1: Technical manuals used for system evaluation

In total, the four manuals comprise 2,848 pages and resulted in 2,813 indexed text chunks stored in the ChromaDB vector database. The number of indexed chunks does not directly correspond to the number of PDF pages, since the documents undergo text extraction and preprocessing before indexing. Consequently, pages containing limited or non-extractable textual content may not result in a corresponding indexed chunk.

The manuals also differ in their structure, length, terminology, and technical content. The ECDIS manual primarily contains information related to navigation-system operation, configuration, alerts, route monitoring, and target-tracking functions. In contrast, the engineering manuals contain information concerning equipment operation, inspection, maintenance procedures, operating limits, safety requirements, and equipment-specific technical instructions.

All four PDF documents were processed using the document-processing and indexing pipeline described in Chapter 2. Text was extracted from the PDF documents and processed prior to embedding generation and storage in the ChromaDB vector database. The source document name and page number were retained as metadata for each indexed chunk, allowing retrieved information to be traced back to its location in the original technical documentation.

The manuals differ considerably in size and technical content. The MFC Valve manual has only six pages, compared with 999 pages for the ECDIS manual and 1,787 pages for the Boiler manual. This combination of technical documents provides a varied basis for evaluating both information retrieval and answer generation. The differences between the manuals in terms of size, terminology, structure, and technical domain make it possible to examine the behavior of the system under different types of technical queries.

3.3 Evaluation Question Design

The evaluation questions were developed on the basis of information explicitly contained in the selected technical manuals. A total of 24 questions were included in the evaluation, with six questions corresponding to each of the four manuals. Using the same number of questions for each manual ensured that all four documents were represented equally in the evaluation, despite their considerable differences in size and number of indexed chunks.

The questions were distributed across the different technical domains represented by the manuals, including navigation, electrical, mechanical, and boiler-related technical information. This approach was adopted to avoid restricting the evaluation to a single type of technical documentation and to examine the system using questions from different areas of maritime operation and engineering.

The evaluation set was designed to include different types of technical queries. These include questions concerning operating procedures, maintenance requirements and intervals, operating limits and conditions, alarm conditions, equipment behavior, safety requirements, and actions required from an operator. The questions also vary in the form of the expected information. Some require the retrieval of a specific numerical value or operating condition, while others require procedural or descriptive information contained in the manuals.

This variation was included to examine the system under different information-retrieval requirements. Questions requesting a specific value or condition generally require the identification of a clearly defined piece of information, whereas procedural or descriptive questions may require information contained within a longer technical passage. Including both types of questions allows the evaluation to cover different ways in which technical information may be presented and requested.

No page numbers or answer locations were provided in the queries. Although some questions identified the equipment or manual concerned, retrieval was performed over the complete indexed collection without restricting the search to a specific document.

3.4 Evaluation Criteria

The evaluation criteria were defined to examine the two main stages of the proposed RAG system: information retrieval and answer generation. Although these stages form part of the same question-answering process, evaluating them separately provides a more detailed understanding of system performance. A generated answer may be inaccurate because the required information was not successfully retrieved, or because the language model did not correctly use information that was available in the retrieved context. Similar distinctions between retrieval quality and generation quality are also considered in RAG evaluation methodologies [15].

3.4.1 Retrieval Evaluation

For each evaluation question, the retrieval component returned the six document chunks with the highest semantic similarity to the submitted query. From each retrieved chunk, the most relevant textual region was subsequently selected using the context-extraction procedure described in Chapter 2 and included in the context provided to the language model. The retrieval evaluation therefore examined whether the information required to answer each question was present within the retrieved context made available to the language model.

A retrieval was classified as successful when at least one of the six retrieved context passages contained sufficient information to answer the question. When none of the retrieved context passages contained the required information, the retrieval was classified as unsuccessful. The assessment was based on the actual content of the retrieved passages rather than semantic similarity scores alone, since a passage may contain terminology related to the query without containing the specific information required to answer it.

In addition to retrieval success, the retrieval rank of the chunk from which the relevant context passage was extracted was recorded. The retrieved chunks are ordered according to their semantic similarity to the query, with Rank 1 representing the highest-ranked result and Rank 6 the lowest-ranked result included in the retrieved context. Recording this position provides additional information about retrieval performance. For example, a relevant passage appearing at Rank 1 indicates that the retrieval mechanism prioritized the required information, whereas a relevant passage at Rank 5 or Rank 6 indicates that the required information was found but ranked below other, less relevant passages.

Six chunks are retrieved for each question, following the configuration used by the implemented system. This value was not selected through a separate optimization experiment. A

retrieval is therefore considered successful when the information needed to answer the question is present within the context passages extracted from these six retrieved chunks and provided to the language model.

3.4.2 Answer Quality Evaluation

The second part of the evaluation focused on the quality of the final answer generated by the language model. For each question, the generated response was compared with the expected information previously identified in the corresponding technical manual.

The purpose of this comparison was not to require an exact reproduction of the wording used in the manual, but to determine whether the generated response preserved the correct technical meaning and adequately answered the question. The generated responses were classified into three categories: Fully Correct, Partially Correct, and Incorrect.

A Fully Correct response accurately provided the information required by the question and preserved the technical meaning of the source documentation. Differences in wording or sentence structure were accepted when they did not alter the meaning of the information. A response could therefore be considered fully correct even when the language model paraphrased the relevant passage from the manual.

A Partially Correct response contained relevant and technically correct information but did not completely satisfy the requirements of the question. This classification was used when important information was omitted, when a procedure or explanation was incomplete, or when additional content introduced ambiguity without making the complete response incorrect.

An Incorrect response was one that failed to provide the information required by the question, provided information inconsistent with the corresponding technical manual, or answered a different aspect of the retrieved technical content without addressing the actual query.

The answer-quality assessment was performed independently from the retrieval classification. This distinction is important because successful retrieval does not necessarily guarantee a fully correct generated answer. The relevant passage may be available in the retrieved context while the language model omits part of the information or interprets it incorrectly.

Conversely, a generated response may occasionally appear technically plausible even when the required information is not present in the retrieved context, for example if the language model does not fully adhere to the instruction to rely exclusively on the provided documentation. In such cases, apparent answer correctness alone does not demonstrate successful retrieval or grounding in the indexed manuals. RAG evaluation approaches similarly distinguish between the relevance of retrieved information and the extent to which the generated response is supported by that information [15].

3.5 Experimental Results

The experimental results are presented using the 24 evaluation questions described in Section 3.3. For each question, the retrieval outcome and the quality of the generated answer were recorded separately according to the criteria defined in Section 3.4. This allows the performance of the retrieval component and the response-generation component to be examined independently as well as in combination.

The results are organized according to the four technical manuals used in the evaluation: the Generator Manual, the ECDIS Instruction Manual, the MFC Valve Maintenance Manual, and the S3106 Boiler Manual. For each test case, the evaluation records whether the required information was successfully retrieved, the rank of the relevant retrieved context when available, and the quality classification of the final generated response.

This structure makes it possible to identify different types of system behavior. A question may result in successful retrieval and a fully correct answer, successful retrieval but an incomplete or inaccurate answer, or unsuccessful retrieval followed by an incorrect or unsupported response. The results therefore provide a basis for examining not only the overall success of the system but also the interaction between retrieval quality and answer generation.

After presenting the results for the individual manuals, the overall performance of the system is summarized using aggregate measures such as retrieval success rate and the distribution of Fully Correct, Partially Correct, and Incorrect responses across the complete evaluation set.

3.5.1 Generator Manual Results

The first group of evaluation questions was based on the Generator Manual and consisted of six questions addressing different types of technical information, including operating parameters, inspection intervals, temperature limits, and maintenance-related information. Table 3.2 presents the retrieval and answer-quality results obtained for the six Generator Manual questions.

Question	Topic	Retrieval	Retrieval Rank	Answer Quality
G1	Generator terminal voltage adjustment range	Unsuccessful	-	Incorrect
G2	Inspection interval during continuous operation	Successful	1	Fully Correct
G3	Bearing trip temperature	Successful	2	Fully Correct
G4	Action at critical insulation resistance	Successful	1	Fully Correct
G5	Recommended oil changing intervals	Successful	1	Partially Correct
G6	Interruption of the excitation circuit during operation	Unsuccessful	-	Partially Correct

Table 3.2: Evaluation Results for the Generator Manual

The Generator Manual achieved successful retrieval for four of the six evaluation questions, corresponding to a retrieval rate of 66.7%. Three generated responses were classified as Fully Correct, two as Partially Correct, and one as Incorrect. The results therefore included both retrieval failures and cases in which the required information was successfully retrieved but the generated response was not fully satisfactory.

For G2 and G4, the information required to answer the questions was contained in the highest-ranked retrieved passage. In G2, the retrieved context correctly identified the inspection interval for continuous operation as approximately 16,000 operating hours or two years, while in G4 it provided the required action when the winding insulation resistance reaches or falls below the calculated critical value. Both questions resulted in Fully Correct responses. G3 also produced a Fully Correct response, although the relevant information was found at Rank 2 rather than Rank 1. In this case, the retrieved passage contained the 100°C trip value for the bearing temperature. These results show that relevant information was not always contained in the highest-ranked retrieval result.

G5 represents a different type of outcome. The required information was successfully retrieved at Rank 1, where the recommended oil change intervals were given as approximately 3,000 and 6,000 operating hours for intermittent and continuous duty, respectively. The generated response correctly provided these values but also introduced unrelated information from the Boiler Manual and additional unnecessary references. The response was therefore classified as Partially

Correct despite successful retrieval. This case illustrates that the availability of the required information in the retrieved context does not necessarily result in a fully correct generated response.

Retrieval was unsuccessful for G1 and G6. For G1, none of the six retrieved context passages contained the required percentage for the continuous adjustment range of the generator terminal voltage. The system returned the predefined insufficient-information response, which was classified as Incorrect with respect to the expected answer. In G6, the retrieved passages contained information related to the excitation system and voltage regulation but did not contain the specific reason required to answer why the excitation circuit must not be interrupted during operation. The generated response nevertheless produced a plausible explanation based on the available context and was classified as Partially Correct.

Overall, the Generator Manual produced the lowest retrieval rate among the four manuals evaluated in this study. The individual cases also illustrate different sources of error within the RAG pipeline: G1 and G6 were associated with retrieval failure, whereas G5 shows that answer quality can still be reduced even when the required information is successfully retrieved. At the same time, the result for G3 shows that relevant information can remain available to the generation stage even when it is not assigned the highest retrieval rank.

3.5.2 ECDIS Manual Results

The second group of evaluation questions was based on the ECDIS Instruction Manual and consisted of six questions concerning different aspects of ECDIS operation, including alert conditions, operator actions, collision-related settings, AIS target capacity, and route-planning functions. As with the Generator Manual evaluation, retrieval performance and answer quality were assessed separately for each question. Table 3.3 presents the results obtained for the six ECDIS evaluation questions.

Question	Topic	Retrieval	Retrieval Rank	Answer Quality
E1	CPA/TCPA danger target alarm activation conditions	Successful	2	Fully Correct
E2	Action after "Arrived LAST WPT" alert	Successful	2	Fully Correct
E3	Purpose of setting CPA and TCPA limits	Successful	1	Fully Correct
E4	Parameters used to determine a danger target	Unsuccessful	-	Partially Correct
E5	Activated AIS targets reaching 95% capacity	Successful	1	Partially Correct
E6	Changing the Cross Track Limit (XTL) during route planning	Successful	1	Fully Correct

Table 3.3: Evaluation Results for the ECDIS Manual

The ECDIS Manual results show that the system successfully retrieved the required information for five of the six evaluation questions, corresponding to a retrieval success rate of 83.3%. For questions E3, E5, and E6, the required information was found in the highest-ranked retrieved context, while for E1 and E2 the relevant information appeared at Rank 2. Four of these five successful retrieval cases resulted in Fully Correct generated responses.

For E1, the relevant passage defining the CPA and TCPA conditions for a dangerous target was retrieved at Rank 2, and the generated response correctly identified the conditions under which the danger-target alarm is activated. Similarly, E2 retrieved at Rank 2 the procedure to be followed after the “Arrived LAST WPT” alert is activated, and the generated answer correctly stated that the operator should confirm the alert and use the [ALERT ACK] button to stop it. E3 also produced a Fully Correct response, with the relevant information concerning the purpose of the CPA and TCPA limits retrieved at Rank 1.

Question E4 was the only unsuccessful retrieval case in this group. Although some of the retrieved passages contained related ECDIS settings and terminology, the retrieved context did not explicitly provide the information required to identify the parameters used to determine a danger target. The generated response nevertheless referred to CPA and TCPA, which are relevant to danger-target determination, but also introduced additional parameters such as AIS data, ship dynamic characteristics, and sector filters that were not supported as danger-target determination criteria by the retrieved context. The response was therefore classified as Partially Correct rather than Fully Correct.

Question E5 demonstrates a different type of error. The required information was successfully retrieved at Rank 1, where the manual states that the “AIS ACT (95% Capacity)” condition occurs when the number of activated AIS targets reaches 95% of the permitted capacity. The generated response correctly reported this condition, but additionally stated that the “AIS ACT (MAX Target)” message is also displayed. The retrieved passage distinguishes these two conditions, with the latter corresponding to the maximum number of activated targets rather than the 95% threshold. Since the response contained the required information but also introduced an incorrect additional statement, it was classified as Partially Correct. This case demonstrates that highly ranked and relevant retrieved information does not necessarily guarantee a fully accurate generated response.

Finally, E6 resulted in successful retrieval at Rank 1 and a Fully Correct response. The retrieved context described the procedure for changing the Cross Track Limit (XTL) during route planning, including selection of the XTL, adjustment of its width, and confirmation of the change. The generated response successfully described the procedure and also incorporated the alternative context-menu method contained in the retrieved passages. This result shows that the system was able to generate an appropriate response for a procedural question requiring information distributed across multiple relevant passages.

In terms of answer quality, four of the six responses (66.7%) were classified as Fully Correct and two (33.3%) as Partially Correct, while no responses were classified as Incorrect. The ECDIS results therefore show a relatively high retrieval success rate and generally accurate answer generation. At the same time, E4 and E5 illustrate two distinct limitations: a partially correct response may be generated when retrieval does not provide sufficient information, while an error may also be introduced by the language model even when the required information is available in the highest-ranked retrieved context.

3.5.3 MFC Valve Manual Results

The third group of evaluation questions was based on the MFC Valve Maintenance Manual and consisted of six questions concerning different aspects of maintenance and troubleshooting, including solenoid valve maintenance, purge operation, solenoid coil replacement, sensor servicing, and flow-path cleaning. As in the previous evaluations, retrieval performance and answer quality were assessed separately for each question. Table 3.4 presents the results obtained for the six MFC Valve evaluation questions.

Question	Topic	Retrieval	Retrieval Rank	Answer Quality
M1	Solenoid valve maintenance under normal operating conditions	Successful	1	Fully Correct
M2	Purpose of periodically using the purge function	Successful	1	Fully Correct
M3	Condition requiring solenoid coil replacement	Successful	1	Fully Correct
M4	Action when additional solenoid valve maintenance is required	Successful	1	Partially Correct
M5	Personnel permitted to service the MFC sensor	Successful	2	Fully Correct
M6	Action when contaminants or particles are found in the MFC bypass or inlet filter	Successful	1	Fully Correct

Table 3.4: Evaluation Results for the MFC Valve Manual

The MFC Valve Manual results show that the system successfully retrieved the required information for all six evaluation questions, corresponding to a retrieval success rate of 100%. For five of the six questions, the required information was contained in the highest-ranked retrieved context. The only exception was M5, for which the relevant passage appeared at Rank 2. Despite this lower ranking, the generated response correctly identified that servicing of the MFC sensor must be performed by a qualified Service Center.

Questions M1, M2, and M3 all resulted in successful retrieval at Rank 1 and Fully Correct generated responses. For M1, the system correctly identified that the solenoid valve does not require maintenance under normal operating conditions. M2 also produced a Fully Correct response, correctly explaining that the purge function is used to fully open the control valve and remove contamination from the surfaces of the solenoid valve. For M3, the system successfully identified the condition requiring solenoid coil replacement, namely a breakage at the solenoid coil cable or connector that may cause valve malfunction.

Question M4 represents the only case in this group in which successful retrieval did not result in a Fully Correct response. The relevant passage was retrieved at Rank 1 and stated that, when additional maintenance of the solenoid valve is considered necessary, the valve should not be disassembled and the MKP authorized Service Center should be contacted. The generated response correctly instructed the user to contact the authorized Service Center but omitted the instruction not to disassemble the valve. Since the response contained correct and relevant information but did not include the complete instruction provided by the manual, it was classified as Partially Correct.

For M5, the relevant information was retrieved at Rank 2 rather than Rank 1. The highest-ranked passage originated from the Boiler Manual and was unrelated to the question, while the second-ranked passage from the MFC Valve Manual explicitly stated that customer access to the sensor is not permitted and that servicing must be performed by a qualified Service Center. The generated response correctly used this information and also distinguished between servicing the sensor itself and replacing its filter screen, resulting in a Fully Correct classification. This case

demonstrates that the system can still generate an accurate response when the most relevant information is not the highest-ranked retrieval result, provided that it is included within the retrieved context.

Finally, M6 resulted in successful retrieval at Rank 1 and a Fully Correct response. The retrieved passage stated that contaminants or particles found in the MFC bypass or inlet filter should be addressed by purging the MFC or using ultrasonic cleaning. The generated response correctly provided both actions. It also mentioned the requirement to return the MFC to an authorized factory service center when toxic or corrosive gases have been used. Although this additional information was not directly required by the question, it was supported by the retrieved passage and did not conflict with the required answer.

In terms of answer quality, five of the six responses (83.3%) were classified as Fully Correct and one (16.7%) as Partially Correct, while no responses were classified as Incorrect. The MFC Valve Manual evaluation also resulted in a retrieval success rate of 100%. The results illustrate two different aspects of the interaction between retrieval and generation: M4 demonstrates that complete and highly ranked retrieved information may still be only partially reflected in the generated response, whereas M5 demonstrates that a Fully Correct response can be produced even when the relevant information appears below an unrelated passage in the retrieval ranking.

3.5.4 Boiler Manual Results

The fourth group of evaluation questions was based on the S3106 Boiler Manual and consisted of six questions concerning safety procedures, maintenance intervals, pressurized equipment, burner operation, boiler filling conditions, and start-up procedures. As in the previous evaluations, retrieval performance and answer quality were assessed separately for each question. Table 3.5 presents the results obtained for the six Boiler Manual evaluation questions.

Question	Topic	Retrieval	Retrieval Rank	Answer Quality
B1	Safety actions before service, inspection, or repair work	Successful	1	Partially Correct
B2	Normal maintenance interval for the chemical dosing pump	Successful	3	Fully Correct
B3	Safety actions before working on pressurized equipment or lines	Successful	3	Fully Correct
B4	Purpose of furnace purging before burner ignition	Unsuccessful	-	Incorrect
B5	Temperature difference between boiler and feedwater during filling	Successful	4	Incorrect
B6	Boiler feedwater level before start-up	Successful	1	Fully Correct

Table 3.5: Evaluation Results for the Boiler Manual

For B1, the relevant information was successfully retrieved at Rank 1. The retrieved passage contained several safety instructions applicable before service, inspection, or repair work, including electrical isolation of the equipment, prevention of accidental reactivation of the mains switch, cutting off the fuel supply, and relieving pressure from pressurized equipment and lines. The generated response correctly instructed the user to cut off the fuel supply, close all shut-off valves, and vent all pressure lines. However, it omitted the electrical isolation requirements and did not fully reproduce the complete set of safety actions contained in the retrieved context. The response was therefore classified as Partially Correct.

Questions B2 and B3 both resulted in successful retrieval at Rank 3 and Fully Correct responses. For B2, the system correctly identified the normal maintenance interval for the chemical dosing pump as three months.

Although the two higher-ranked passages were related to the chemical dosing pump, the specific maintenance interval was contained in the third-ranked passage. Similarly, for B3, the first two retrieved passages contained boiler-related information but did not provide the required safety procedure. The relevant passage at Rank 3 stated that pressure must be relieved, shut-off valves closed, and pressure lines vented before work is started on pressurized equipment or lines. The generated response accurately reproduced these required actions.

B4 was the only question for which the required information was not successfully retrieved. The returned context contained several passages related to furnace purging, ignition burners, and burner operating sequences, but none provided the specific explanation required by the question. The generated response subsequently associated furnace purging before ignition with the removal of residual fuel oil from the burner lance and with proper atomization and combustion. This explanation was derived from passages describing related but different purging procedures and did not correctly answer the question concerning the purpose of furnace purging before burner ignition. B4 was therefore classified as both an unsuccessful retrieval case and an incorrect generated response.

For B5, the relevant passage was retrieved at Rank 4. It stated that the boiler must be filled very slowly when the temperature difference between the boiler and the feedwater exceeds approximately 50°C. Although the question asked for a maximum permitted difference, the passage described a condition requiring slow filling rather than an absolute upper limit. The generated response nevertheless presented 50°C as the maximum permitted difference and omitted the slow-filling requirement. It was therefore classified as Incorrect because it reproduced the numerical value but changed its technical meaning.

Finally, B6 resulted in successful retrieval at Rank 1 and a Fully Correct response. The highest-ranked passage explicitly stated that the boiler should be filled with feedwater to approximately 50 mm below the normal water level before start-up. The generated response provided this value directly and without introducing additional unsupported information. In terms of answer quality, three of the six responses (50.0%) were classified as Fully Correct, one (16.7%) as Partially Correct, and two (33.3%) as Incorrect. Retrieval was successful for five of the six questions (83.3%).

The Boiler Manual results show that a correct answer can be generated from information retrieved below Rank 1, as demonstrated by B2 and B3. However, successful retrieval did not always lead to a correct or complete response. B1 omitted required safety information, while B5 misinterpreted the numerical condition stated in the retrieved passage. B4 represents a different case, in which the required information was absent from the retrieved context and the generated explanation was incorrect.

3.5.5 Overall Results

The results from the four technical manuals were combined in order to provide an overall assessment of the performance of the proposed RAG system. In total, 24 evaluation questions were considered, with six questions corresponding to each manual. Table 3.6 summarizes the retrieval success and answer-quality results obtained for each manual and for the complete evaluation set.

Technical Manual	Successful Retrievals	Retrieval Success Rate	Fully Correct	Partially Correct	Incorrect
Generator Manual	4/6	66.7%	3	2	1
ECDIS Manual	5/6	83.3%	4	2	0
MFC Valve Manual	6/6	100%	5	1	0
Boiler Manual	5/6	83.3%	3	1	2
Total	20/24	83.3%	15	6	3

Table 3.6: Overall Evaluation Results

Across the complete evaluation set, the system successfully retrieved the information required to answer 20 of the 24 questions, corresponding to an overall retrieval success rate of 83.3%. Retrieval performance varied between the individual manuals. The MFC Valve Manual achieved successful retrieval for all six questions, while the ECDIS and Boiler manuals each achieved successful retrieval for five of the six questions. The Generator Manual produced the lowest retrieval success rate, with relevant information successfully retrieved for four of the six questions.

In terms of answer quality, 15 of the 24 generated responses (62.5%) were classified as Fully Correct, while six responses (25.0%) were classified as Partially Correct and three responses (12.5%) as Incorrect. The MFC Valve Manual produced the highest number of Fully Correct responses, with five out of six, followed by the ECDIS Manual with four. The Generator and Boiler manuals each produced three Fully Correct responses.

The combined results also show that retrieval success and answer quality were related but did not always correspond directly. In several cases, relevant information was successfully retrieved at positions below Rank 1 and was nevertheless used to produce a Fully Correct response. Examples include G3, E1, E2, and M5 at Rank 2, and B2 and B3 at Rank 3. These results indicate that retrieving multiple candidate passages allowed relevant information to remain available to the generation stage when the highest-ranked passage did not contain the information required to answer the question.

At the same time, successful retrieval did not always result in a Fully Correct answer. In G5, E5, M4, and B1, relevant information was available in the retrieved context, but the generated response was incomplete or included additional content that reduced its accuracy or clarity. These responses were classified as Partially Correct. In B5, the relevant passage was also retrieved, but the generated answer misinterpreted the temperature condition as an absolute upper limit and was classified as Incorrect. These cases demonstrate that retrieval performance alone is not sufficient to determine the quality of the final system response.

Conversely, unsuccessful retrieval did not always result in a completely incorrect response. In G6 and E4, the required information was not present in the retrieved context, but the generated responses still contained technically relevant information and were therefore classified as Partially Correct. In contrast, G1 and B4 demonstrate two different outcomes of unsuccessful retrieval. For G1, the system returned the predefined fallback response rather than providing an unsupported numerical value, whereas in B4 the model generated an incorrect explanation based on retrieved passages concerning related but different purging procedures.

Taken together, the results indicate that the proposed system was able to retrieve relevant technical information for the majority of the evaluation questions, achieving an overall retrieval success rate of 83.3%. Fully Correct responses were produced for 62.5% of the questions, while a further 25.0% were classified as Partially Correct. In total, 21 of the 24 responses (87.5%) were classified as either Fully Correct or Partially Correct. This combined percentage should not be interpreted as an overall accuracy measure, since Partially Correct responses did not fully satisfy the requirements of the corresponding questions. The observed failure cases show that errors can arise both when the required information is absent from the retrieved context and when relevant information is available but is incorrectly interpreted during answer generation.

3.6 Error Analysis and Failure Cases

Although the experimental results presented in Section 3.5 indicate that the system successfully retrieved relevant information for the majority of the evaluation questions, the evaluation also revealed several cases in which the retrieval or response-generation stage did not perform as expected. Examining these cases separately is important for understanding the limitations of the implemented RAG pipeline and for identifying the stage at which each error occurred.

The observed errors can generally be divided into two main categories. The first consists of retrieval failures, in which the information required to answer the question was not present in the retrieved context provided to the language model. The second consists of generation-related errors, in which relevant information was successfully retrieved but the generated response omitted part of the required information, introduced irrelevant or inaccurate content, or did not fully reflect the retrieved evidence. In addition, some retrieval failures resulted in partially correct responses, illustrating that a technically plausible generated answer does not necessarily indicate that the answer was grounded in the retrieved documentation.

Four of the 24 evaluation questions were classified as retrieval failures: G1, G6, E4, and B4. In these cases, none of the six retrieved context passages contained the specific information required to answer the corresponding question. However, the behavior of the response-generation component differed considerably between these cases.

In G1, the question requested the percentage range within which the generator terminal voltage can be continuously adjusted under steady-state conditions. Although several of the retrieved passages were related to generator voltage and voltage regulation, none contained the required $\pm 5\%$ value. As a result, the system returned the predefined fallback response, “Not enough information in the provided manuals.” This represents a retrieval failure, since the required information exists in the indexed manual but was not included in the retrieved context. At the same time, the fallback mechanism prevented the model from generating an unsupported numerical value.

A different behavior was observed in G6 and E4. In both cases, the retrieved context contained information that was semantically related to the question but did not contain the specific information required for a complete answer. For G6, the retrieved passages included information

concerning the excitation system and generator voltage behavior, but did not contain the manual statement that interruption of the excitation circuit during operation can give rise to voltage surges. Nevertheless, the generated response associated interruption of the excitation circuit with an undesirable increase in generator voltage and possible equipment damage. Similarly, in E4, the retrieved passages contained information related to CPA/TCPA settings and AIS functions, but did not provide the specific conditions used to identify a danger target. The generated response correctly referred to CPA and TCPA, but also introduced other parameters that were not part of the required criteria. Both responses were therefore classified as Partially Correct despite unsuccessful retrieval.

The B4 case provides another example of the effect of retrieving related but insufficient information. The retrieved passages contained information associated with burner operation, purging, and ignition, but did not contain the specific explanation required by the question concerning the purpose of sufficiently purging the furnace before burner ignition. The generated response consequently referred to a different purge-related procedure and did not correctly answer the question. This case was therefore classified as both an unsuccessful retrieval and an incorrect response.

In addition to the retrieval failures, five evaluation questions—G5, E5, M4, B1, and B5—showed cases in which the required information was successfully retrieved but the generated response was not fully correct. G5, E5, M4, and B1 were classified as Partially Correct, while B5 was classified as Incorrect. These cases indicate that successful retrieval alone does not guarantee that the retrieved information will be accurately and completely reflected in the generated answer.

In G5, the required information was retrieved at Rank 1 and clearly stated that the recommended oil-changing intervals were approximately 3,000 operating hours for intermittent duty and 6,000 operating hours for continuous duty. The generated response included both correct values, but also introduced irrelevant information from the Boiler Manual and incomplete references to retrieved documents. Although the main technical information was correct, the additional content reduced the clarity and precision of the response, resulting in a Partially Correct classification.

A similar generation-related problem was observed in E5. The Rank 1 retrieved passage explicitly stated that the AIS ACT (95% Capacity) message is displayed when the number of activated AIS targets reaches the 95% threshold. The generated response correctly identified this message but additionally stated that the AIS ACT (MAX Target) message is also displayed. According to the retrieved passage, the latter corresponds to a different condition in which the maximum number of activated AIS targets has been reached. The response therefore combined two distinct alert conditions, despite the required information being available in the highest-ranked retrieved context.

In M4 and B1, the main issue was the omission of relevant information rather than the introduction of incorrect information. For M4, the retrieved context stated that when additional solenoid valve maintenance is required, the valve should not be disassembled and an authorized MKP Service Center should be contacted. The generated response correctly recommended contacting an authorized Service Center but omitted the instruction not to disassemble the valve. Similarly, in B1, the retrieved passage contained several safety actions that should be performed before service, inspection, or repair work, including electrical isolation, prevention of accidental reactivation, interruption of the fuel supply, and depressurization of relevant lines. The generated response included several of these actions but omitted part of the required safety information. Both responses were therefore technically relevant but incomplete.

B5 involved a different type of interpretation error. The relevant passage, retrieved at Rank 4, stated that the boiler must be filled very slowly when the temperature difference between the boiler and the feedwater exceeds approximately 50°C. The generated answer instead presented 50°C as the maximum permitted difference. Although the question itself asked for a maximum permitted value, this interpretation was not supported by the retrieved passage. The response was therefore classified as Incorrect because it changed the meaning of the numerical condition.

These cases demonstrate that a successful retrieval can still be followed by errors or omissions in the generated response. Even when the required evidence is present in the retrieved context, the generation stage may omit relevant details, combine information from different retrieved passages, or misinterpret an operating condition. This distinction is particularly important in technical documentation, where an incomplete maintenance instruction, safety procedure, or operating condition may be significant even when the remaining information is technically correct.

Taken together, these failure cases highlight the importance of evaluating retrieval and answer generation as separate components of the RAG pipeline. Retrieval failures did not always result in the same generation behavior: the system could return the predefined fallback response, as in G1, or generate a technically plausible but only partially grounded or incorrect response from semantically related context, as observed in G6, E4, and B4. Conversely, G5, E5, M4, B1, and B5 demonstrate that successful retrieval does not necessarily guarantee a fully correct response, since relevant information may still be omitted, combined with unrelated information, or inaccurately represented during answer generation.

The results also show that relevant information was sometimes available below the highest-ranked retrieval result. Several Fully Correct responses were produced when the relevant information appeared below Rank 1, including G3, E1, E2, and M5 at Rank 2, and B2 and B3 at Rank 3. These cases show that relevant information remained available to the generation stage even when the required passage was not assigned the highest semantic similarity score.

3.7 Discussion

The experimental results provide an indication of the extent to which the proposed locally deployed RAG system can support information retrieval and question answering over maritime technical documentation. Across the 24 evaluation questions, the system achieved a retrieval success rate of 83.3%, while 62.5% of the generated responses were classified as Fully Correct and a further 25.0% as Partially Correct. These results suggest that the implemented approach was able to retrieve relevant technical information and generate useful responses in the majority of the evaluated cases, although the difference between retrieval success and fully correct answer generation also demonstrates that locating the required information does not necessarily guarantee that it will be completely and accurately represented in the final response.

This distinction is particularly relevant to the architecture of the proposed system. The retrieval component and the language model perform different functions within the RAG pipeline, and the evaluation showed that performance at either stage can affect the final result. In several cases, the required information was successfully retrieved, but the generated response remained incomplete or introduced additional information. Conversely, some retrieval failures produced technically plausible responses even though the specific supporting information was absent from the retrieved context. Consequently, the quality of a RAG response should not be assessed solely on whether it appears technically reasonable; the availability and relevance of the supporting retrieved information are also important when evaluating whether the response is grounded in the source documentation.

An important aspect of the proposed system is that both retrieval and response generation are performed locally. The system uses locally stored technical documentation, ChromaDB for vector storage and retrieval, and Qwen2.5-3B-Instruct for response generation without relying on an external cloud based language model service. Similar approaches have demonstrated the use of local RAG architectures for document retrieval and question answering, in which document processing, retrieval, and language-model inference can be performed locally rather than through external cloud-based services [16]. The experimental results of the present study indicate that a relatively compact language model can likewise be integrated into a local RAG architecture and provide useful responses to technical questions across different types of maritime documentation.

Local execution is particularly relevant to the intended maritime application of the system. Previous research has identified uneven and unreliable digital connectivity in shipboard environments [17]. A locally deployed architecture reduces dependence on continuous external connectivity by allowing the indexed manuals and the question-answering pipeline to remain available on the host computer without requiring each query to be transmitted to an external language model service. The evaluation conducted in this study did not directly examine network availability, data privacy, or operational reliability at sea; therefore, these characteristics should be considered design considerations of the local architecture rather than experimentally validated outcomes of the present evaluation.

The results also show some variation in retrieval performance across the four technical manuals. The MFC Valve Manual achieved successful retrieval for all six evaluation questions, while the corresponding retrieval success rates were 83.3% for both the ECDIS and Boiler manuals and 66.7% for the Generator Manual. These differences occurred despite substantial variation in document size, from the six-page MFC Valve Manual to the 1,787-page Boiler Manual. The results therefore indicate that relevant information could be retrieved from both small and large documents within the indexed collection.

However, the present evaluation does not provide sufficient evidence to establish a direct relationship between document size and retrieval performance. Only six questions were evaluated for each manual, and retrieval effectiveness may also depend on factors such as the wording of the query, the terminology used in the source document, the distribution of related information across the manual, and the semantic similarity between relevant and non-relevant passages. The higher retrieval performance observed for one manual should therefore not be interpreted as evidence that smaller documents are inherently easier for the system to process or retrieve from.

The retrieval-rank results provide further insight into the behavior of the system. Although relevant information was frequently retrieved at Rank 1, several Fully Correct responses were generated from cases in which the required information appeared at a lower position in the retrieved results. This suggests that restricting the generation context to only the highest-ranked passage would not necessarily have been sufficient for all of the evaluated questions. By providing the language model with relevant textual regions selected from the top six retrieved chunks, the implemented pipeline allowed information from lower-ranked results to remain available during answer generation.

At the same time, the results suggest a potential trade-off when multiple passages are included in the generation context. While a broader context increases the possibility that the required information will be available to the language model, it may also introduce passages that are only partially related or irrelevant to the specific question. The behavior observed in some Partially Correct responses indicates that additional retrieved information can occasionally influence the generated response even when the correct evidence is already present. Therefore, the results suggest that retrieval performance depends not only on whether relevant information is included within the retrieved set, but also on the relevance and quality of the additional context presented alongside it.

The experimental results should nevertheless be interpreted within the scope of the evaluation conducted in this study. The evaluation consisted of 24 questions distributed equally across four technical manuals. Although the questions covered different types of technical information, including operating conditions, maintenance requirements, safety procedures, alarms, and numerical values, the evaluation set represents only a limited sample of the information contained in the indexed documentation. The reported percentages therefore describe the performance of the implemented system on the specific evaluation set and should not be interpreted as a general accuracy estimate for all maritime technical manuals or all possible technical questions.

Furthermore, the evaluation focused primarily on retrieval success and the technical correctness of the generated responses. Other characteristics that may affect the practical use of a locally deployed RAG system, such as response time, computational requirements, user interaction, and performance with additional document types, were not systematically evaluated in the present experiments. These aspects therefore remain outside the scope of the reported results and provide possible directions for further evaluation.

Overall, the findings support the feasibility of using a locally deployed RAG architecture to support question answering over maritime technical documentation. The system successfully retrieved the required information for the majority of the evaluated questions and was able to generate technically useful responses across different maritime technical domains. At the same time, the observed retrieval and generation errors show that access to relevant documentation alone does not eliminate the possibility of incomplete, inaccurate, or insufficiently grounded responses. The results therefore support the potential of local RAG systems for facilitating access to technical information, while also highlighting the importance of maintaining traceability to the original documentation. Generated responses should therefore be considered a means of supporting the location and interpretation of technical information rather than a replacement for the authoritative source manuals.

Chapter 4 – Conclusion and Future Work

4.1 Summary of the Work

This thesis presented the design, implementation, and evaluation of a locally deployed Retrieval-Augmented Generation (RAG) system for question answering over maritime technical manuals. The system was developed to provide users with direct access to relevant technical information through natural-language questions, offering an alternative to manually searching through extensive technical documentation. Both information retrieval and response generation are performed locally, without relying on external language model APIs.

The implemented architecture combines document processing, semantic retrieval, and local language-model generation. Maritime technical manuals are processed page by page and divided into overlapping text chunks, which are converted into vector representations using the all-MiniLM-L6-v2 SentenceTransformer model. The resulting embeddings are stored in a persistent ChromaDB vector database together with source metadata. When a question is submitted, the same embedding model is used to represent the query and retrieve the six most semantically similar document chunks. A relevant textual passage is then selected from each retrieved chunk and incorporated into the context provided to the language model.

Response generation is performed locally using Qwen2.5-3B-Instruct. The model is instructed to answer using the information provided in the retrieved manual context and to return a predefined response when sufficient information is not available. The system also retains information about the source manual and page associated with the retrieved content, allowing the supporting documentation to be traced alongside the generated response. The complete pipeline therefore operates locally, from document indexing and retrieval to language-model inference. A graphical user interface was also developed to provide a more direct way for users to submit technical questions and view the generated responses together with the corresponding source information.

The proposed system was experimentally evaluated using four maritime technical manuals that differ substantially in size and content. A total of 24 evaluation questions were used, with six questions constructed for each manual. Retrieval performance and generated-answer quality were assessed separately in order to distinguish between cases in which the required information was not successfully retrieved and cases in which relevant information was available but was not completely or accurately reflected in the generated response. This evaluation provided the basis for examining the performance and limitations of the proposed architecture.

4.2 Main Findings

The experimental evaluation showed that the proposed system was able to retrieve the information required to answer the majority of the evaluation questions. Successful retrieval was achieved for 20 of the 24 questions, corresponding to an overall retrieval rate of 83.3%. Retrieval performance varied across the four technical manuals, ranging from 66.7% for the Generator Manual to 100% for the MFC Valve Manual, while both the ECDIS and Boiler manuals achieved a retrieval rate of 83.3%.

The generated responses were classified as Fully Correct for 15 of the 24 questions (62.5%), Partially Correct for six questions (25.0%), and Incorrect for three questions (12.5%). These results show that successful retrieval did not always result in a fully correct response. In several cases, the required information was available in the retrieved context, but the generated answer omitted relevant details, included additional content, or misinterpreted the retrieved information. This distinction highlights the importance of evaluating the retrieval and generation stages separately when examining the performance of a RAG system.

The evaluation also showed that the most relevant information was not always contained in the highest-ranked retrieval result. Several Fully Correct responses relied on information found at Rank 2 or Rank 3. In B5, the relevant passage was retrieved at Rank 4, but the generated response misinterpreted its meaning and was classified as Incorrect. These results show that the position of a retrieved passage and the correctness of the generated answer are distinct aspects of system performance.

Performance also differed between manuals with substantially different characteristics and sizes. The MFC Valve Manual achieved successful retrieval for all six questions and produced five Fully Correct responses, whereas the Generator Manual achieved the lowest retrieval rate among the four manuals. At the same time, the ECDIS and Boiler manuals, despite being considerably larger, both achieved successful retrieval for five of their six evaluation questions. Within the scope of this evaluation, these results do not indicate a direct relationship between document size and retrieval performance, which may also depend on factors such as the relationship between the wording of a question and the indexed document content.

Taken together, the findings indicate that the lightweight local RAG architecture was capable of retrieving relevant technical information and generating useful responses for the majority of the evaluated questions across different maritime technical manuals. However, the observed retrieval failures and partially correct responses also show that access to relevant document content does not by itself guarantee a complete and accurate generated answer. The results therefore highlight the importance of both retrieval quality and response generation in determining the final output of the system.

4.3 Limitations

The evaluation conducted in this thesis has several limitations that should be considered when interpreting the results. First, the experimental dataset was limited to four maritime technical manuals and 24 evaluation questions. Although the selected manuals differed considerably in size and technical content, the evaluation represents only a limited sample of the wide range of documentation encountered in maritime applications. The results should therefore not be interpreted as representative of all types of maritime technical manuals or technical questions.

A second limitation concerns the evaluation methodology. Retrieval success and answer quality were assessed manually based on the information contained in the technical manuals and the generated responses. While this approach allowed individual retrieval and generation cases to be examined in detail, the evaluation was conducted on a relatively small set of questions and did not include multiple independent evaluators. Consequently, the classification of responses into Fully Correct, Partially Correct, and Incorrect categories may involve a degree of subjective judgement.

The experiments were also conducted using a single system configuration. The proposed architecture used all-MiniLM-L6-v2 for semantic embeddings, ChromaDB for vector retrieval, a Top-6 retrieval configuration, and Qwen2.5-3B-Instruct for response generation. Alternative embedding models, language models, retrieval depths, chunking configurations, or retrieval strategies were not systematically compared. The experimental results therefore characterize the implemented configuration rather than establishing that the selected components or parameter settings are optimal for maritime technical question answering.

Another limitation is that the implemented document-processing pipeline operates on extracted textual content. Technical manuals can contain tables, diagrams, drawings, and other visual information that may be important for answering certain technical questions. The present system does not include a dedicated multimodal processing mechanism for interpreting such information. Its ability to answer questions whose required evidence is primarily contained in visual or highly structured document elements is therefore limited by the available text extraction and retrieval process.

Finally, the system was developed and evaluated in a local computer environment rather than under operational shipboard conditions. The experiments demonstrate that the complete retrieval and generation pipeline can be executed locally on the hardware used in this study, but they do not evaluate its performance under actual vessel operating conditions. Factors such as operational workload, hardware constraints on different onboard computers, long-term system reliability, and practical interaction with maritime personnel were outside the scope of the experimental evaluation.

4.4 Future Work

Future work could extend the evaluation of the proposed system to a larger and more diverse collection of maritime technical documentation. Including additional types of equipment manuals and a larger set of evaluation questions would provide a broader basis for assessing the performance of the system across different technical domains. The use of multiple independent evaluators could also strengthen the assessment of generated-answer quality and reduce the influence of subjective judgement in the evaluation process.

Further experimentation could investigate alternative configurations of the retrieval and generation pipeline. Different embedding models, language models, chunk sizes, chunk overlaps, and numbers of retrieved passages could be systematically compared in order to examine their effect on retrieval and answer quality. Alternative retrieval approaches, such as hybrid retrieval combining semantic and keyword-based methods or the use of a reranking stage, could also be investigated.

Another direction for future development is the processing of information that is not adequately represented through extracted text alone. Technical manuals frequently contain tables, diagrams, drawings, and other visual elements that may contain information required to answer technical questions. Integrating multimodal document-processing capabilities could allow future versions of the system to retrieve and interpret both textual and visual information from technical documentation.

The generation stage could also be further investigated to improve the consistency of responses with the retrieved context. The experimental results showed cases in which the required information was successfully retrieved but the generated response remained partially correct because relevant information was omitted or additional unrelated content was introduced. Future work could therefore examine alternative prompting strategies, context-construction methods, or language models to reduce these types of generation errors while maintaining local execution.

Another possible extension would be the addition of persistent conversation history. In the current implementation, questions and answers remain visible only during the active session. A future version could store previous conversations and allow users to reopen and continue earlier interactions through the graphical interface. This could make the system more practical for repeated consultation of technical manuals and for reviewing information retrieved during previous sessions.

A further practical improvement would be to package the system as a standalone desktop application. The current version is executed through the Python development environment, whereas a packaged version could allow users to launch the system directly without interacting with development tools. Separate builds could be created for macOS and Windows, making the system easier to distribute and use in different local computing environments.

Finally, the system could be evaluated in an operational shipboard environment. Such an evaluation would allow its performance to be examined on actual onboard computing hardware and under realistic operating conditions. It could also investigate practical aspects that were outside the scope of the present study, including interaction with maritime personnel, long-term system operation, hardware constraints on different onboard computers, and the suitability of the local RAG architecture for routine access to technical documentation onboard vessels.

In conclusion, this thesis shows that semantic retrieval can be combined with a compact locally deployed language model to support question answering over maritime technical manuals. The experimental results demonstrate that the proposed architecture can provide useful access to technical information while also revealing limitations in both retrieval and response generation. Although further development and broader evaluation are required, the implemented system provides a practical foundation for the continued investigation of locally deployed RAG-based tools for accessing maritime technical documentation.

Chapter 5 – Bibliography

- [1] A. Vaswani et al., “Attention Is All You Need,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [2] D. Rothman, *RAG-Driven Generative AI: Build Custom Retrieval Augmented Generation Pipelines with LlamaIndex, Deep Lake, and Pinecone*. Birmingham, U.K.: Packt Publishing, 2024.
- [3] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [4] L.-F. Bouchard and L. Peters, *Building LLMs for Production: Enhancing LLM Abilities and Reliability with Prompting, Fine-Tuning, and RAG*. Towards AI, 2024.
- [5] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.
- [6] S. Soman and S. Roychowdhury, “Observations on Building RAG Systems for Technical Documents,” *ICLR 2024 Tiny Papers Track*, 2024.
- [7] Y. Pu, Z. He, T. Qiu, H. Wu, and B. Yu, “Customized Retrieval Augmented Generation and Benchmarking for EDA Tool Documentation QA,” *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2024.
- [8] B. E. Galvan et al., “A Multi-Agent LLM Framework with RAG for Shipbuilding Engineering Documents,” *Proceedings of the 18th International Conference on Agents and Artificial Intelligence (ICAART)*, 2026.
- [9] P. E. Doherty, “Shipboard Operating and Maintenance Procedures and the Knowledge Gap,” *Journal of Marine Engineering & Technology*, 2016.
- [10] S. Nam et al., “LoRA-Tuned Multimodal RAG System for Technical Manual QA: A Case Study on Hyundai Staria,” *Applied Sciences*, 2025.
- [11] L. Rasaq, M. Siddula, O. P. Yadav, R. Walthall, J. Ensberg, and P. Yadav, “Edge-Deployed Generative Language-Based Retrieval for Aerospace Asset Health Management,” *European Conference of the Prognostics and Health Management Society*, 2026.
- [12] J. Wulf, J. Meierhofer, and N. Pascher, “Guided Operations and Maintenance with Retrieval-Augmented Generation: System Design and Evaluation at a Machine Manufacturer,” *Procedia CIRP*, 2026.
- [13] I.-S. Han, M.-I. Roh, and M.-C. Kong, “A RAG-based Q&A System for Ship Regulations Applying Domain Adaptation,” *International Journal of Naval Architecture and Ocean Engineering*, 2026.
- [14] Y. Ren, M. Chen, J. Weng, S. Zhang, X. Xiao, and K. Liu, “Deploying Efficient LLM Agents on Maritime Autonomous Surface Ships: Fine-Tuning, RAG, and Function Calling in a Mid-Size Model,” *Information*, 2026.
- [15] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, “RAGAs: Automated Evaluation of Retrieval Augmented Generation,” *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 150–158, 2024.
- [16] M. Waqas, “cainsearch: A privacy-first hybrid search and local RAG system for personal document collections,” *SoftwareX*, 2026.
- [17] R. B. Jensen, “Fragmented digital connectivity and security at sea,” *Marine Policy*, 2021.
- [18] W. X. Zhao et al., “A Survey of Large Language Models,” *Frontiers of Computer Science*, 2026.
- [19] J. Wei et al., “Finetuned Language Models Are Zero-Shot Learners,” *International Conference on Learning Representations (ICLR)*, 2022.

- [20] A. Yang et al., “Qwen2.5 Technical Report,” arXiv, 2024.
- [21] L. Huang et al., “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions,” ACM Transactions on Information Systems, 2025.
- [22] I. Yadav, S. Schindler, D. Peters, and R. Klinger, “External Knowledge Integration in Large Language Models: A Survey on Methods, Challenges, and Future Directions,” Semantic Web, 2026.
- [23] P. Zhao et al., “Retrieval-Augmented Generation for AI-Generated Content: A Survey,” Data Science and Engineering, 2026.

Appendix A – Evaluation Questions and Reference Information

This appendix presents the 24 questions used in the experimental evaluation of the proposed RAG system. For each question, the reference information used for the manual assessment of answer quality is provided together with the corresponding source manual. The reference information was identified directly from the technical documentation and represents the information required for a response to correctly address each evaluation question.

A.1 Generator Manual Evaluation Questions

Question	Evaluation Question	Reference Information	Source
G1	According to the generator manual, within what percentage of the rated voltage can the generator terminal voltage be continuously adjusted under steady-state conditions?	The generator terminal voltage can be continuously adjusted within $\pm 5\%$ of the rated voltage under steady-state conditions.	Generator Manual, p. 23
G2	How often should the generator be inspected during continuous operation?	The basic inspection interval for continuous operation is approximately 16,000 operating hours or 2 years, whichever occurs first.	Generator Manual, p. 7
G3	At what bearing temperature should the generator be stopped immediately?	The bearing safety-device trip value is 100°C . The manual also states that the machine should be stopped immediately if the bearing temperature exceeds the normal operating value by 15 K.	Generator Manual, p. 32
G4	What should be done if the generator winding insulation resistance reaches or falls below the calculated critical value?	The windings must be dried; the manual indicates that the rotor may need to be removed as part of the drying procedure.	Generator Manual, pp. 4–5
G5	What are the recommended lubricating oil change intervals for the generator bearings under intermittent and continuous duty?	The recommended oil-changing intervals are approximately 3,000 operating hours for intermittent duty and 6,000 operating hours for continuous duty.	Generator Manual, p. 29
G6	Why must the excitation circuit of the generator not be interrupted during operation?	Interruption of the excitation circuit during operation can cause voltage surges and must therefore be avoided.	Generator Manual, p. 23

Table A.1: Evaluation Questions and Reference Information for the Generator Manual

A.2 ECDIS Manual Evaluation Questions

Question	Evaluation Question	Reference Information	Source
E1	Under what condition is the ECDIS danger target alarm (CPA/TCPA) activated?	The danger-target alarm is activated when $CPA \leq CPA \text{ Limit}$ and $0 \leq TCPA \leq TCPA \text{ Limit}$. All of these conditions must be met for the alarm to be issued.	ECDIS Manual, p. 320
E2	What action should be taken when the "Arrived LAST WPT" alert is activated?	After confirming the content of the alert, the operator should click the [ALERT ACK] button to acknowledge and stop the alert.	ECDIS Manual, p. 345
E3	What is the purpose of setting the CPA Limit and TCPA Limit in the ECDIS?	The CPA and TCPA limits define the collision-detection conditions used for collision-avoidance alert generation. Their values should be selected according to the ship type, sea area, weather, and marine conditions.	ECDIS Manual, p. 670
E4	What parameters are used by the ECDIS to determine a danger target?	A target is classified as a danger target based on CPA and TCPA in relation to the configured CPA Limit and TCPA Limit.	ECDIS Manual, p. 320
E5	What happens when the ECDIS detects that the number of activated AIS targets has reached 95% of its capacity?	The "AIS ACT (95% Capacity)" condition is generated when the number of activated AIS targets reaches 95% of the permitted capacity (tolerance).	ECDIS Manual, p. 319
E6	How can the Cross Track Limit (XTL) be changed during route planning?	Select the XTL, move the cursor to adjust its width, and click the mouse button to confirm the change. The XTL can also be changed through the [Change XTL] context-menu function.	ECDIS Manual, p. 446

Table A.2: Evaluation Questions and Reference Information for the ECDIS Manual

A.3 MFC Valve Manual Evaluation Questions

Question	Evaluation Question	Reference Information	Source
M1	Does the solenoid valve of the MFC require maintenance under normal operating conditions?	The solenoid valve requires no maintenance under normal operating conditions, although periodic use of the purge function is necessary.	MFC Valve Maintenance Manual, p. 4
M2	What is the purpose of periodically using the purge function on the MFC solenoid valve?	The purge function completely opens the control valve and is used to remove contamination adhering to surfaces in the solenoid valve.	MFC Valve Maintenance Manual, p. 4
M3	Under what condition is the solenoid coil replacement procedure required?	The replacement procedure is required when there is a breakage at the solenoid coil cable or connector, which can cause valve malfunction.	MFC Valve Maintenance Manual, p. 4
M4	What should be done if additional maintenance of the MFC solenoid valve is considered necessary?	The valve should not be disassembled. An MKP authorized Service Center should be contacted.	MFC Valve Maintenance Manual, p. 4
M5	Who is permitted to perform service on the MFC sensor?	Customer access to the sensor is not permitted. Any service on the sensor must be performed by a qualified Service Center. The filter screen, however, may be replaced by the customer.	MFC Valve Maintenance Manual, p. 3
M6	What should be done if contaminants or particles are found in the MFC bypass or inlet filter?	The MFC should be purged or cleaned using ultrasonic cleaning. If the MFC has been used with a toxic or corrosive gas, it must be returned to an authorized factory service center for repairs.	MFC Valve Maintenance Manual, p. 2

Table A.3: Evaluation Questions and Reference Information for the MFC Valve Manual

A.4 Boiler Manual Evaluation Questions

Question	Evaluation Question	Reference Information	Source
B1	What should be done before starting service, inspection, or repair work on the boiler equipment?	The equipment must be electrically isolated, the mains switch must be secured against accidental reactivation, and the fuel supply must be cut off.	S3106 Boiler Manual, p. 482
B2	What is the normal maintenance interval for the chemical dosing pump?	The normal maintenance interval is three months, based on approximately 30% operating time. The interval should be shortened in cases of heavier use, such as continuous operation.	S3106 Boiler Manual, p. 1335
B3	What should be done before starting work on pressurized boiler equipment or lines?	Relieve the pressure, close all shut-off valves, and vent all pressure lines before starting work.	S3106 Boiler Manual, p. 482
B4	Why must the furnace be sufficiently purged before burner ignition?	The furnace must be sufficiently purged because insufficient purging may cause a furnace explosion.	S3106 Boiler Manual, p. 744
B5	What is the maximum permitted temperature difference between the boiler and the feedwater when filling the boiler?	If the temperature difference between the boiler and feedwater exceeds approximately 50°C, the boiler must be filled very slowly.	S3106 Boiler Manual, p. 496
B6	To what level should the boiler be filled with feedwater before start-up?	The boiler should be filled with feedwater to approximately 50 mm below the normal water level.	S3106 Boiler Manual, p. 496

Table A.4: Evaluation Questions and Reference Information for the Boiler Manual