Enhancing Autonomous Ship Communication: A Cost-Effective and High-Accuracy LLM Framework Using Decision Trees and RAG

Jaebin Ku
Department of Computer Engineering
Chungnam National University
Daejeon, Republic of Korea
0009-0001-1694-9670

Sanha Kim
Department of Computer Engineering
Chungnam National University
Daejeon, Republic of Korea
0009-0003-4615-3045

Eunkyu Lee
Autonomous Ship Research Center
Samsung Heavy Industries
Daejeon, Republic of Korea
0009-0000-4903-5288

Umar Zaman
Department of Computer Engineering
Chungnam National University
Daejeon, Republic of Korea
0000-0002-4257-4868

Kyungsup Kim
Department of Computer Engineering
Chungnam National University
Daejeon, Republic of Korea
0000-0002-0166-1439

***Abstract*—This study introduces a novel architecture designed to enhance the performance and cost-efficiency of Large Language Models (LLMs) in autonomous ship communication systems. Autonomous ships require high accuracy and rapid response, yet their operational constraints—limited computational resources and lack of internet connectivity—pose significant challenges for traditional LLMs. To address these issues, we integrate Retrieval-Augmented Generation (RAG) with Decision Trees. This integration improves the efficiency of RAG's data retrieval and processing, significantly reduces the computational overhead of LLMs, and enhances response accuracy. Evaluations using 200 hours of real-world maritime communication data demonstrate that the proposed system outperforms existing methods in speed and accuracy under resource-constrained conditions. This research advances the practical application and reliability of LLMs in autonomous ship communication, providing a strong foundation for improving automation and ensuring safety in the maritime industry.**

***Index Terms*—LLM, RAG, Decision Tree, Communication System, Autonomous ship,**

I. Introduction

Recent advancements in LLMs have demonstrated remarkable achievements in the field of natural language processing. The high performance and flexibility of LLMs have the potential to assist or replace human operators across various industrial domains [1]. Similarly, the maritime industry has begun exploring the adoption of LLM systems, particularly for autonomous ships.

Autonomous ships, which can perceive their surroundings, plan routes, and avoid hazards with minimal human intervention, are emerging as a cornerstone of next-generation maritime. According to the International Maritime Organization's (IMO) MASS (Maritime Autonomous Surface Ship) trial guidelines, autonomous ship technology is classified into four levels based on the degree of human involvement. At Level 1, all operational decisions are made by humans onboard, with automated systems providing support through route suggestions and situational awareness. Level 2 allows ships to support significant aspects of autonomous operation, but critical decisions, such as communication and accident response, remain under human control. Level 3 envisions operations where human involvement on the bridge is entirely optional under normal conditions, although remote monitoring and control are required to handle abnormal situations. Level 4 represents full autonomy, where ships can operate independently in all scenarios without human intervention. At present, autonomous ship technology is confined to Level 2. To elevate their operational capability to Level 3, companies worldwide are actively conducting research in various areas. Among these, the development of automated VHF radio communication systems for autonomous ships has been frequently highlighted as a pivotal technology for advancing their operational autonomy [2]–[5].

Integrating LLMs into the maritime domain poses significant challenges. Chief among these are issues related to the unique spatial characteristics of the maritime environment, including the high costs of deployment, hallucination errors, and the generation of unstructured responses with unnecessary information. At sea, internet access is unreliable, and managing physical resources is difficult. As such, systems like the Global Maritime Distress and Safety System (GMDSS), critical for autonomous ships, must operate without internet connectivity, and this includes communication devices. Additionally, onboard computer systems rely on industrial-grade hardware designed for durability and reliability, but these systems generally have lower CPU and GPU performance compared to computers used in land-based research. Reliable and standalone LLM systems require substantial computa-

tional resources to ensure accurate performance. In the case of VHF radio communication, which is essential for safe navigation, route adjustments, and collision avoidance, quick and reliable responses are critical. However, building an LLM system capable of meeting these demands can cost tens of thousands to millions of dollars, even on land. Reducing the cost by scaling down computational resources may lead to increased hallucination issues, which in turn can compromise system reliability. To mitigate hallucinations, LLMs often include excessive information in their responses, exacerbating the problem of unstructured responses.

In summary, the automation of VHF radio communication for autonomous ships, a system is required that can function reliably without internet access, deliver fast and accurate responses, generate well-structured responses, and operate efficiently with minimal computational resources at a low cost. Such systems are expected to integrate components like communication interfaces, STT (Speech-to-Text), TTS (Text-to-Speech), and LLMs. However, LLMs face significant challenges due to their resource demands and the constraints of the maritime domain. One promising approach to addressing these challenges is RAG, which combines LLMs with information retrieval systems. RAG enhances reliability by providing domain-specific information before the response generation process, thus improving the accuracy and relevance of LLM outputs. This approach has demonstrated its capability to mitigate issues such as hallucination and high costs in various applications, suggesting that it holds promise for addressing similar challenges in the maritime domain.

However, the maritime domain differs fundamentally from other domains due to its unique spatial constraints. While RAG can alleviate some issues, such as hallucinations and cost, it does not inherently ensure structurally appropriate responses. Unstructured responses often occur when LLMs include excessive information in their outputs to avoid hallucinations. This issue is particularly pronounced in smaller LLMs, which are often necessary for resource-constrained environments like autonomous ships. Smaller models are more likely to generate unstructured responses due to their limited capacity to balance response accuracy and computational efficiency. These limitations in computational resources could potentially be mitigated through fine-tuning. However, the maritime domain poses additional challenges. For example, VHF radio communication data is often restricted due to security concerns, making it difficult to obtain sufficient data for effective fine-tuning. Even if such data were available, the small size of models suitable for deployment on autonomous ships limits the benefits of fine-tuning. Currently, among LLMs that meet the requirements of the autonomous ship domain, the Llama 3.1 model with 8B parameters is one of the few options.

Therefore, in this study, we propose a novel architecture that combines RAG with Decision Trees to enhance RAG's performance and prevent unstructured responses. Decision Tree-based language models represent one of the most traditional forms of language modeling. While they lack the ability to generate responses autonomously, they excel at selecting predefined responses, making them highly applicable in various industrial domains. The responses from Decision Trees can be guided and enforced by their designers, making them particularly effective for state-specific information responses. In contrast to LLMs, which are probabilistic language models that generate responses by predicting context-based continuations, Decision Tree-based language models rely on selecting predefined responses for given inputs, functioning more as response systems rather than generators. This characteristic ensures that Decision Tree-based models do not produce unstructured responses. The following sections will explain and compare the operational mechanisms of RAG and Decision Trees. Interestingly, RAG's information retrieval process shares similarities with the response selection process of Decision Trees, suggesting that their integration is both logical and effective. By combining RAG with Decision Trees, the resulting system can retrieve information necessary for generating responses while the response selection process is guided by predefined rules set by the designer. This allows the system to actively participate in response construction of LLMs, blending the attributes of a response system and a sentence generator. Consequently, the integrated model addresses the issue of unstructured responses effectively. And the low computational cost and high accuracy of Decision Trees will ensure fast response times and high precision, even when used with smaller LLM models. Conversely, RAG and LLM will support the response diversity of Decision Trees. Ultimately, the research outcomes introduced in this study will provide core technologies for automating VHF radio communication systems in the domain of autonomous ships.

II. Theoretical background

A. LLM:response generation principle

An LLM is one of the most significant innovations in natural language processing, capable of learning from vast amounts of text data to understand and generate patterns, contexts, and relationships between words. LLMs operate based on the Transformer architecture and Self-Attention mechanisms [6], [7]. The Transformer structure processes input data in parallel while learning relationships between words, thereby efficiently preserving contextual information. This process consists of two stages: Pre-training and Fine-tuning. In the Pre-training stage, the model learns general language patterns using an unsupervised learning approach, while the Fine-tuning stage involves additional training with domain-specific knowledge, such as translating environmental inputs or performing question-and-answer tasks [8].

The operational principle of an LLM involves analyzing input sentences and sequentially predicting the most likely word tokens to generate a response. This method ensures linguistic continuity, making the model appear capable of human-level language generation based on its training data. However, such functionality requires training on extensive datasets, resulting in large model sizes and significant computational resource consumption. Additionally, when presented with input that has low relevance to the data learned during the pre-training

stage, LLMs are prone to generating incorrect or unsupported answers, a phenomenon known as *hallucination*.

B. RAG:response generation principle for LLM

RAG is a technology designed to expand the limited knowledge of LLMs by combining their language generation capabilities with information retrieval techniques to enhance the quality of responses. The operational principle of RAG consists of two main stages. The first stage is the retrieval phase, where relevant information is searched from an external database based on the input question [9]. Techniques like DPR (Dense Passage Retriever) are used in this process, transforming both the question and the documents stored in the database into vectors for similarity comparison [10]. For languages like Korean, Japanese, and English, embedding techniques such as morphological analysis and one-hot encoding are commonly employed to convert sentences into vectors. In essence, this process involves comparing the regions in the multidimensional space pointed to by respective sets of morphemes. The second is the generation phase, where the LLM generates a response based on the retrieved information. During this process, RAG uses the retrieved information as *context*, and the LLM creates sentences based on this context. The context refers to all vector groups within the predefined *context length* range specified by the designer, originating from the region indicated by the question vector. Depending on the question, multiple contexts can be selected, and their inclusion can be mutually exclusive based on the design.

The primary role of RAG is to expand and complement the knowledge of LLMs. RAG enables LLMs to retrieve and utilize information from external knowledge databases, supplementing what the LLM has internally learned [11]. This capability allows RAG to provide updates or domain-specific data that LLMs cannot independently access. Additionally, whereas LLMs operate solely on their fixed training data, RAG can retrieve information from databases or the web in real-time, making it suitable for dynamic queries. As a result, with the support of RAG, LLMs can utilize external knowledge sources without the need to create larger, more complex models. Instead of developing an enormous LLM containing all possible knowledge, combining a core language model with RAG allows the construction of smaller, yet powerful systems. This demonstrates that RAG can simultaneously achieve lightweight models and improved reliability. Furthermore, as RAG enables LLMs to use retrieved information, it provides clear justifications for responses and mitigates the issue of hallucinated responses.

However, as mentioned earlier, the maritime domain is unique compared to other domains due to its spatial and operational constraints. The maritime environment is characterized by limited internet access and challenges in resource supply and management. Consequently, all onboard systems must operate offline, relying on durable and reliable industrial-grade computers for computing resources. These constraints necessitate the use of minimized LLM models, such as the Llama 3.1 8B model, which has a size of approximately 34.07GB. Even with RAG, such models may not completely resolve hallucination issues. Similarly, fine-tuning is unlikely to yield significant improvements due to the limited computational resources available to these smaller models. Moreover, since the computational cost of response generation rests entirely on the LLM, the issue of unstructured responses will persist.

Nevertheless, RAG remains a highly effective solution for enhancing LLM performance. With advancements in its components, RAG's capabilities continue to improve. Key developments include the evolution of DPR, which enables efficient and accurate information retrieval, the extension of multimodal capabilities to retrieve and generate not only text but also other data types such as images and videos, and the optimization of knowledge databases. The Decision Tree-based architecture proposed in this paper also contributes to this progress. By integrating Decision Trees with RAG, the architecture allows RAG to select intended contexts through a Decision Tree-optimized database and directly participate in the construction of LLM responses. This approach addresses the issue of unstructured responses and ensures more reliable and trustworthy responses.

C. Decision Tree:response generation principle for Single-Turn Interaction

The Decision Tree is a rule-based structural approach that uses a tree-like data structure to navigate predefined paths based on the given input and determine the appropriate response. The operation of a Decision Tree is straightforward: input data starts at the root node of the tree, and at each node, specified conditions (if-else) are evaluated to move down to the subsequent nodes. Ultimately, the process reaches a leaf node, where the response defined at that node is returned. For example, the tree might check whether the user's input contains specific keywords and select a corresponding path based on the result. This approach allows developers to define clear rules, ensuring high reliability and predictability in structured dialogues or tasks. In this study, the Decision Tree is applied to single-turn interaction systems, similar to those supported by general LLMs. Single-turn interaction refers to a conversational model where a clear and concise response is provided for a single user input without continuing the conversation. This model is particularly useful in environments where immediate and rapid responses are required. Applying a Decision Tree to single-turn interactions ensures that each step and response in the dialogue can be explicitly defined, guaranteeing high reliability. However, this approach lacks flexibility, making it challenging to handle unexpected inputs or dynamically incorporate new information.

This response selection method is highly similar to that of RAG. Each node in a single-turn interaction Decision Tree represents morphemes that may be included in the question. In other words, when a question traverses the entire Decision Tree and reaches a leaf node, it indicates that all relevant morphemes have been compared and analyzed for response selection. This process closely resembles RAG's morpheme vector comparison and analysis. The key difference lies in how

weights are assigned: while RAG assigns equal weight to all vector components and compares similarity probabilistically, the Decision Tree halts further comparisons if the condition fails at an upper node. This characteristic highlights the Decision Tree's lack of flexibility compared to RAG but also implies that morphemes in upper nodes are more significant, reflecting the hierarchical dependency of a sentence's structure. If Decision Tree and RAG were combined—allowing RAG to assign different weights to each vector component and Decision Tree to gain flexibility in response selection—their respective shortcomings could be mitigated. Such a system would enable the creation of a more precise and effective response generation framework.

III. Method

A. The relationship between Decision Tree and RAG

This study aims to enhance the performance of LLMs by integrating Decision Tree with RAG. Before explaining the methodology, this study will first examine the relationship between Decision Tree and RAG based on the operational process of RAG.

To discuss this, it is essential first to understand the response selection mechanism of RAG in detail. RAG begins its process by structurally preparing training data and an external knowledge base during the initial system setup. Various knowledge sources, such as documents, databases, web materials, research papers, and dictionaries, are collected and divided into appropriately sized text passages as defined by the designer. For languages like Korean, Japanese, and English, this process can involve morphological analysis algorithms. The system then converts these segmented text passages into vectors (embeddings) and stores them in a multidimensional space. For example, a document containing the sentence "Please avoid the ship port-to-port" might be processed using a morphological analysis algorithm, splitting it into components such as "please," "avoid," "the," "ship," "port," "to," "port," and so on. These components are then transformed into vectors and stored in the multidimensional space. When a user input is received, the system compares and analyzes the input vector with the pre-stored vectors using the same principles. Similarity algorithms such as cosine similarity or Euclidean distance measurements are commonly employed for this comparison.

Assuming that $\mathbf{Vector\ P} = (p_1, p_2, p_3, \ldots, p_n)$ and $\mathbf{Vector\ Q} = (q_1, q_2, q_3, \ldots, q_n)$ exist in a multidimensional space (p_i and q_i are embedding integers of morphemes.):

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}} \tag{1}$$

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{2}$$

The vector search process of RAG closely resembles the node selection process of a Decision Tree chatbot. While Decision Tree chatbots can be designed in various forms depending on the developer's approach, in single-turn interactions like those handled by LLMs, the most common method involves checking whether the user's input contains specific morphemes through conditional statements. Such Decision Trees are typically configured to achieve higher accuracy by assigning nodes with morphemes of varying importance at each depth level, based on real conversation records or empirical evidence. When a user provides input, the input is segmented into morphemes, and the Decision Tree is traversed. At each depth level, the presence of morphemes is checked repeatedly until the final response is determined.

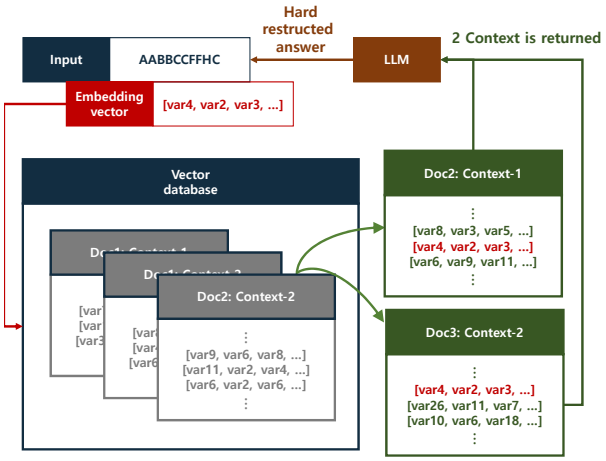

Fig. 1: RAG that refer to common documents

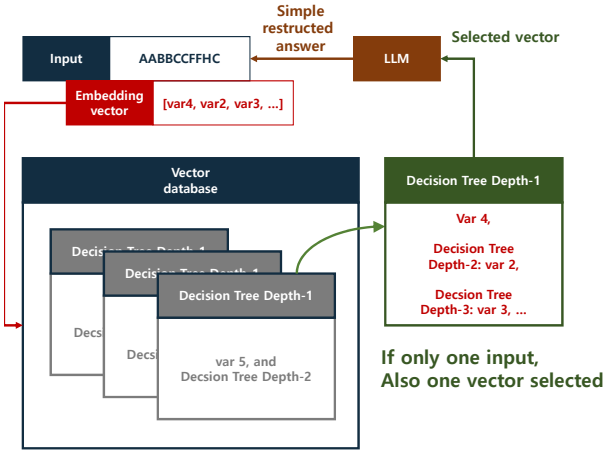

Fig. 2: RAG that refer to Decision Tree documents

The critical difference between RAG and a Decision Tree lies in the prioritization of morphemes during the vector selection process. In a Decision Tree, each morpheme has a distinct priority, whereas RAG uses vectors composed of morpheme sets with uniform priority. RAG ensures flexibility in response selection by matching the user's input vector to the most similar document vector stored in the multidimensional

space. However, if the multidimensional space contains multiple vectors with similar morphemes but different meanings, or if the vector dimensions are insufficient, RAG may fail to guarantee accuracy.

Furthermore, while a Decision Tree directly uses the data stored in the leaf node as the response, RAG selects surrounding documents based on the embedded input vector within the context length and assists the LLM in reconstructing the response. While this operation of RAG can partially mitigate issues such as LLM hallucination and computational costs, it does not directly participate in the response generation process. As a result, when the computational resources of the LLM are insufficient, RAG cannot fully guarantee the reliability of the responses.

The issue of computational resources for LLMs becomes even more pronounced in scenarios where extremely small models, such as those required in the domain addressed in this study, must be used. If the domain has very limited data, making fine-tuning impossible, and is also closed off, leaving the LLM with little to no knowledge related to the expected responses for user inputs, the information provided by RAG could potentially create confusion. This is due to the inherent characteristic of LLMs to "answer comprehensively." Specifically, LLMs operate by interpreting inputs based on context, but when presented with entirely unfamiliar contexts or methods of interpretation, and without any information falling within the range of comprehensive response generation, probabilistic selection might lead the LLM to distrust and disregard the information retrieved by RAG. This could, in turn, result in new hallucinations or contribute to the generation of unstructured responses.

Therefore, this study proposes a novel architecture where RAG references a Decision Tree as its source of information. Traditional RAGs, which refer to general documents or databases, convert all sentences or content within a specific information set into vectors with equal priority, selecting the context region corresponding to the input vector as the reference document for generating a response. In contrast, when RAG references a Decision Tree, each sentence or piece of content corresponds to individual morphemes related to specific inputs or responses. As a result, the selected context region comprises a set of morphemes. In the former case, the LLM generates a response that probabilistically incorporates the selected context region, while in the latter case, the LLM can directly use the final node of the Decision Tree—pointed to by the selected morpheme set—as the response. This final node represents a response guided by the designer for the given input. Thus, the LLM transitions from functioning solely as a sentence generator to also possessing the properties of a response system. Consequently, when RAG references a Decision Tree, the information selected as the context region and provided to the LLM is significantly reduced, lowering the computational cost for the LLM. This allows RAG to more effectively address the hallucination and computational cost issues associated with LLMs, enabling even extremely small-scale models to maintain reliability. Furthermore, the issue of unstructured responses is also resolved.

B. Data Processing and Decision Tree Design

This study reflects the characteristics of the target domain in designing the Decision Tree by processing and establishing design criteria based on the features of Korean sentence structures and maritime communication data. The research involved converting 200 hours of VHF radio communication records from coastal vessels operating in South Korea into text. These records were analyzed and classified into 103 primary categories by maritime navigation experts. This classification was conducted to accurately reflect common communication patterns and critical communication information required during maritime operations.

During the data processing phase, text data was analyzed at the morpheme level using the Kkma morpheme analyzer from the KoNLPy library. The text was segmented into components such as subject (S), modifier (M1), adverbial phrase (M2), predicate (P), and object (O) based on the standard word order in Korean. Semantic priorities were assigned within each sentence. For instance, in maritime communication, the clear identification of the sender and receiver is critical, which is why the subject was set as the top node in the Decision Tree [12].

In the design of the Decision Tree, hierarchical semantic feature descriptions were applied to construct the structure of the nodes [13], [14]. Each node was composed of morpheme units, with essential components of Korean sentences distinguished from supplementary information and assigned to higher-level and lower-level nodes, respectively. Specifically, each element was evaluated for its substantive meaning and relational meaning through semantic feature analysis. Substantive meaning refers to the critical elements that determine the core meaning of a sentence, while relational meaning supports these substantive elements. Since Korean sentences are interpreted around predicates, substantive and relational elements play a crucial role in guiding the interpretation of the predicate.

The structure of the designed Decision Tree is shown in Figure 3. This design aims to maximize the accuracy of system responses in the maritime navigation domain. In particular, the criteria for node traversal are based on the importance and relevance of sentence components, optimizing the analysis of user input to deliver accurate responses. The study also accounted for variability in sentence patterns to design a structure that can flexibly handle communication errors or irregular inputs. This flexibility is achieved through elements such as endings or particles commonly used in the Korean language. As a result, the Decision Tree effectively processes complex sentence structures and provides highly accurate responses. In conclusion, the RAG referencing the Decision Tree as a preprocessing document plays a crucial role in automating VHF radio communication for ships by generating reliable responses with minimal computational resources.

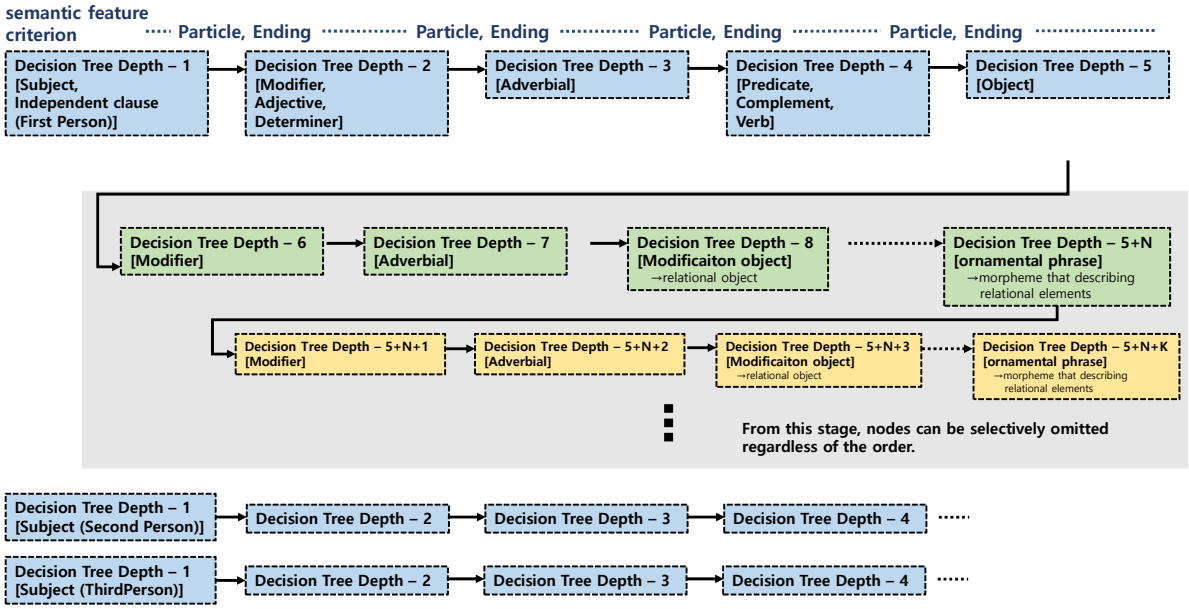

Fig. 3: Decision Tree Design by Layer Semantic feature descriptions

C. Overall System Configuration

The final system design incorporates a Decision Tree, RAG, and LLM. At a high level, the principle of determining output based on user input remains the same as the conventional RAG-based LLM system. However, the type of data stored and the format in which context is returned after RAG translates and creates vectors from reference documents has been modified. Unlike the conventional architecture where the LLM needed to reconstruct responses within the context range provided by RAG, the updated architecture determines the response group during the RAG process itself. This allows the LLM to provide clear and efficient responses. The LLM's role is simply integrating the pre-determined response group using simple connectors, such as conjunctions.

While integrating the Decision Tree with RAG might appear to limit the LLM's flexibility in responses, the RAG does not select a context if there is no relevant information in the vector space for the given input. Additionally, the Decision Tree contains only the essential semantic components required for each conversation type, minimizing the risk of selecting incorrect contexts due to excessive documents. Even though the Decision Tree fixes RAG's response types, the LLM retains the flexibility to support free-form conversations for unexpected conversational flows, as long as the Decision Tree has not explicitly addressed them. This flexibility can also be maintained in intended conversational flows, depending on how the Decision Tree's prompts are configured during RAG's design phase. As a result, the Decision Tree, RAG, and LLM complement one another, preserving their respective strengths while addressing each other's weaknesses.

RESULT

The desktop system used to obtain the results in this study comprised an AMD Ryzen 7 9700X processor, a GeForce RTX 4080 SUPER 16GB graphics card, and four DDR5 PC5-44800 32GB memory modules. This setup represents a typical workstation used for office and research purposes. The dataset utilized in the research consisted of 200 hours of actual radio communication dialogue recorded during maritime vessel operations. When converted into text, the actual dialogue exchanges, excluding silent intervals, amounted to fewer than 1,000 instances. This reflects the typical characteristics of domains with limited data frequency and restricted public availability due to security concerns. Thus, the dataset was highly suitable for validating the research goal of achieving high performance with constrained data and resources.

Evaluating the performance of LLMs is inherently complex and challenging to define. Sentence interpretation can vary depending on individual perspectives, making it difficult to establish clear evaluation criteria. Traditional metrics like Translated Error Rate (TER) or Character Error Rate (CER) are often used to assess language model performance but are not suitable for LLM research. Specifically, in this study, where the RAG system utilizes a Decision Tree to retrieve response groups stored in the leaf nodes, these metrics are not reliable. Instead, the most commonly used evaluation method for such studies is human evaluation, particularly by domain experts. While this approach does not standardize answers into fixed rules, resulting in longer evaluation periods and potentially varying levels of reliability depending on the individual evaluator, it remains the most trustworthy method currently available. In this study, five experts actively working

Model name	Llama3.1-8B (None Decision Tree)	Llama3.1-70B (None Decision Tree)	Llama3.1-405B -> GPT4o (None Decision Tree)	Llama3.1-8B (with Decision Tree)
Overall system size (before lightening)	34.07GB	290.8GB	820GB	34.07GB
Overall system size (after lightening)	7.6 GB	42.9GB	231.9GB	7.6 GB
RAM Usage	7.2GB	40.2GB	210GB	7.2GB
Execute time (for one response)	within 5 seconds	More than 40 seconds, less than 60 seconds	Can not running Llama3.1 in setted system. GPT-4o case, within 5 seconds	within 5 seconds
Accuracy	20 out of 200 correct answers	110 out of 200 correct answers	180 out of 200 correct answers	170 out of 200 correct answers

Fig. 4: Llama3.1 Results by Model Size

in the field of maritime navigation research participated in the evaluation to calculate accuracy rates. If two or more of the five experts raised objections to a response, it was deemed incorrect. The criteria for correctness included whether the response adequately addressed the input, the structural appropriateness of the response for the domain, and the ease of understanding the contextual meaning. The evaluations were primarily conducted alongside simulation monitoring, though some were performed onboard autonomous ships operating in the test navigation zones of Geoje Island waters. The results showed an accuracy rate of 80% to 90% based on the lightweight Llama 3.1 8B model (approximately 7.6GB). Detailed evaluations of resource consumption, execution speed, and accuracy for each model are provided in Figure 4.

IV. Conclusion

This study proposed and validated a novel architecture that combines Decision Tree and RAG to enhance the performance of a domain-specific LLM for autonomous ships. This architecture is designed to ensure high accuracy and structured responses in domains where only minimal LLMs can be used due to the unique spatial constraints of the maritime environment, which lacks internet support and access to high-performance computing resources. The study demonstrated that the Decision Tree-based RAG system can provide accurate answers even when using small-scale LLMs.

The proposed architecture was designed based on 200 hours of voice data recorded in actual ship operating environments, and its performance was evaluated by experts actively working in the field. This process demonstrated that the system is practical and applicable in real-world environments.

Ultimately, this study proposes a method to develop LLMs into more effective and reliable tools within the domain of autonomous ships. It decisively addresses the high-cost issues of LLMs that previous research has struggled to resolve, while ensuring practical applicability and reliability in real-world environments through structured responses.

V. Acknowledgment

This work was partly supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-00155857, Artificial Intelligence Convergence Innovation Human Resources Development (Chungnam National University) and ”Regional Innovation Strategy (RIS)” through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE)(2021RIS-004).

References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx et al., “On the Opportunities and Risks of Foundation Models.” arXiv, 2021.

[2] D. Pei, J. He, K. Liu, M. Chen, and S. Zhang, “Application of Large Language Models and Assessment of Their Ship-Handling Theory Knowledge and Skills for Connected Maritime Autonomous Surface Ships,” Mathematics, vol. 12, no. 15. MDPI AG, p. 2381, 2024.

[3] L. Chen and J. Liu, “Identification of Shipborne VHF Radio Based on Deep Learning with Feature Extraction,” Journal of Marine Science and Engineering, vol. 12, no. 5. MDPI AG, p. 810, 2024.

[4] E. C. Nakilcioglu, M. Reimann, and O. John, “Adaptation and Optimization of Automatic Speech Recognition (ASR) for the Maritime Domain in the Field of VHF Communication,” arXiv, 2023.

[5] A. Haq and M. Suryanegara, “Speech Recognition System Using DeepSpeech Architecture Method on VHF Radio Communication for Tanker Ship Officers at Indonesian Sea Ports,” 2022 9th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). IEEE, pp. 164–168, 2022.

[6] A. Vaswani, “Attention is all you need,” Advances in Neural Information Processing Systems, 2017.

[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal et al., “Language Models are Few-Shot Learners.” arXiv, 2020.

[8] A. Radford, “Improving language understanding by generative pre-training,”, 2018.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” arXiv, 2020.

[10] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov et al., “Dense Passage Retrieval for Open-Domain Question Answering.” arXiv, 2020.

[11] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, “REALM: Retrieval-Augmented Language Model Pre-Training.” arXiv, 2020.

[12] Y. Kim and C. Ok, ”Korean Semantic Role of subcategorization,” in Proceedings of the 26th Conference on Hangul and Korean Language Information Processing, pp.143-148, 2014

[13] C. Lee, “A Study on Structures of Semantic Features according to the Analysing Levels of Korean Verbs,” The Society of Korean Semantics, vol. 8, pp. 133-170, 2001

[14] J. Sin and C. Ok, ”Semantic resources and word sense disambiguation for Korean semantic analysis,” Communications of the Korean Institute of Information Scientists and Engineers, vol. 34, no. 8, pp. 8–16, 2016.