An On-premises Agentic Retrieval-Augmented Generation Framework for Interrogating Nuclear Safety Documents

Bokyeong Kim[ab], Seoyeon Jung[a], Minseok Ko [a], SoYoung Kim[a], YunBum Park [a], Yonggyun Yu[ab*]
[a]*Korea Atomic Energy Research Institute, 111, Daedeok-daero 989 beon-gil , Yuseong-gu , Daejeon, 34057, Korea*
[b]*Korea National University of Science & Technology, 217, Gajeong-ro, Yuseong-gu , Daejeon, 34113, Korea*
[*]*Corresponding author: ygyu@kaeri.re.kr*

Keywords : Agentic RAG, nuclear safety documents, FSAR analysis, on-premises deployment, retrieval-augmented generation,

1. Introduction

Recent generative Artificial Intelligence (AI) technology has been revolutionizing the paradigm of knowledge production and information utilization across various industries. In the nuclear industry, while core design information is comprehensively documented in extensive technical documents such as the Final Safety Analysis Report (FSAR), practical tasks such as export proposals and technical response documents require comprehensive analysis of highly specialized information from multiple domains including mechanical, core, and safety analysis. Due to these complex requirements, there are practical limitations for a small team of specialized personnel to master all technical information spanning tens of thousands of pages and respond promptly to inquiries.

To address these challenges, there has been growing interest in intelligent document analysis systems utilizing Retrieval-Augmented Generation (RAG) technology. However, the nuclear field has unique characteristics where the use of external network-based commercial AI services is restricted due to stringent security requirements, and the risk of critical technical information leakage must be fundamentally prevented. Furthermore, conventional simple RAG approaches show limitations in generating comprehensive responses to complex technical documents due to their single retrieval-generation pattern and lack of multi-step reasoning capabilities.

Therefore, this study aims to develop an intelligent question-answering system that provides reliable answers by analyzing FSAR documents through an agentic approach that mimics the reasoning process of human experts in a secure on-premises environment. The proposed Agentic RAG system maximizes answer reliability and transparency by systematically decomposing complex questions into sub-goals, iteratively performing information retrieval and reasoning, and clearly tracking the source sections, tables, and figures that serve as the basis for answers.

In this study, we conducted comprehensive experiments in both API-based and on-premises environments to validate the performance of the proposed system. We constructed an evaluation dataset comprising expert-reviewed queries that reflect the complexity of real-world usage scenarios, and established the conventional Vanilla RAG system as our baseline to confirm the practical applicability in on-premises environments through performance comparisons.

2. Methods and Experiments

The system proposed in this study is designed to deeply understand large-scale nuclear engineering documents and generate highly reliable answers to complex user queries. For this research, we utilized ‘Chapter 1: Introduction and General Description of the Plant’ and ‘Chapter 5: Reactor Coolant System and Connecting Systems’ from NuScale's publicly available Final Safety Analysis Report (FSAR).

2.1 Agentic RAG Architecture

Inspired by the process of a human expert who formulates plans and verifies information in stages to solve complex problems, we have adopted an agentic architecture that autonomously decomposes goals and solves them sequentially. This structure is managed by states based on the LangGraph library and operates in a four-step cyclical process, as shown in Figure 1.

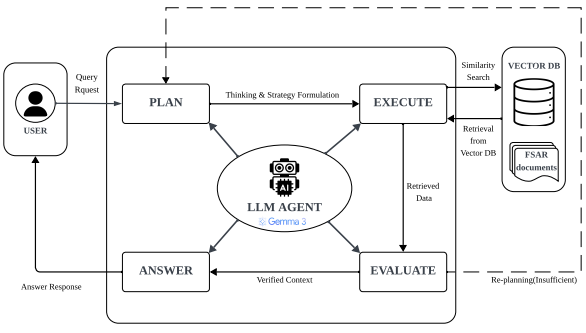

Fig.1. Operational Flow of the LLM Agent for FSAR Document Analysis

- Plan Agent: When a user's question is input, the agent designs an overall path to reach the final answer and formulates a plan for the first step. In this stage, it predicts the most valid information

sources by leveraging the structural features of the SSAR document.

- Execute Agent: Following the established plan, the agent uses a tool to retrieve relevant information from the vector-indexed SSAR documents and generates an intermediate answer to the sub-question based on that information.
- Evaluate Agent: The agent self-evaluates whether the answer is sufficient based on the execution results. If it determines that more information is needed, it formulates a new plan for the next step, incorporating the previous results, and repeats the execution.

2.2. Domain-Specific Prompt Design for Nuclear Engineering

The performance of each agent step is determined by the instructions provided to the LLM, namely the prompts. In this study, prompts were designed considering the characteristics of nuclear documents. In the 'Plan' stage, the agent was instructed to establish analysis pathways by leveraging the structural features of FSAR documents (sections, tables, figures), while in the 'Execute' stage, it was directed to generate answers based solely on retrieved information, cite sources for all claims, and quote technical data values and units verbatim from the original text. Table 1 shows examples of such prompts.

Table I: Prompt Engineering Templates for Each Agent Step

Agent	System Role	Prompt Template
PLAN	Multi-step Reasoning Planner	"You are a multi-step reasoning planner for nuclear FSAR documents. Design a reasoning pathway leveraging FSAR structure (sections, tables, figures). Create an anchor checklist with Section (§), Table, and Figure IDs. Formulate one precise search query targeting FSAR anchors. Always follow the document-first, no-assumption principle." *Example Query: "Table 4.1-1" OR "design parameter" OR "control rod drop time" OR "§4.2.1.5"*
EXECUTE	FSAR Information Analyst	"You are an FSAR information analyst. Use only the retrieved text/tables/figures as evidence. For every statement, attach citations in the form [SOURCE N | FSAR: Section/Table/Figure]. Quote technical values and units verbatim from the original text. If conflicting values exist, report both explicitly and highlight the uncertainty. If evidence is insufficient, declare 'insufficient evidence' instead of guessing." *Example Query: [SOURCE 2 | Table 4.1-1] shows reactor pressure as 2250 psia(15.5 MPa)*
EVALUATE	Completeness Assessor	"You are a completeness assessor. Evaluate whether the gathered evidence sufficiently answers the question. Score completeness from 0.0–1.0. Consider (a) parameter coverage, (b) source quality and FSAR anchoring, (c) conflict resolution, and (d) operational context alignment. Continue reasoning if score < 0.8, otherwise proceed to final answer."

2.3 Experimental Query Processing Example

Figure 2 demonstrates the results of the Agentic RAG system developed in this study processing water chemistry control queries from actual NuScale FSAR documents. When a user inputs the query "What are the reactor coolant water chemistry controls for NuScale SMR?", the system retrieves and provides detailed chemical concentration limits including chloride (≤ 0.15 ppm), fluoride (≤ 0.15 ppm), dissolved oxygen (≤ 0.005 ppm), sulfate (≤ 0.15 ppm), hydrogen, and boron concentrations. The system provides precise source citations for each chemical parameter in the format [SOURCE 2 | FSAR: Table 5.2-5], and all technical data are quoted verbatim from the original text with their original values and units.

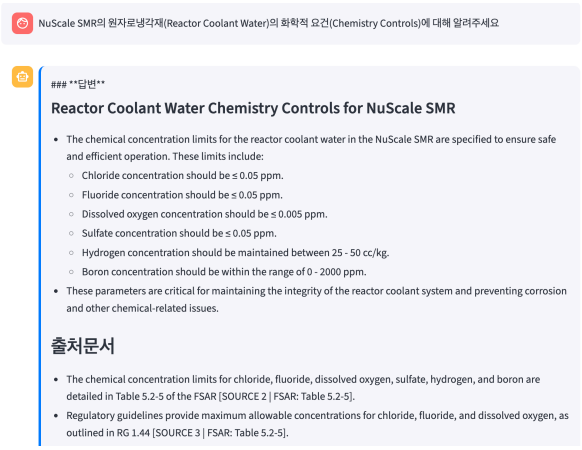

Fig. 2. Example Output of Agentic RAG System for NuScale Reactor Vessel Design Parameters Query.

3. Evaluation

In this study, we conducted experiments in two environments to validate the performance of the proposed Agentic RAG system. The first environment was an API-based setup utilizing OpenAI GPT-4o and the text-embedding-large model, while the second was an on-premises setup combining the Gemma-3 27B Instruction model with the BGE-M3 embedding model.

The evaluation dataset consisted of 10 expert-reviewed queries provided by the nuclear safety analysis division. Although the dataset size is limited,

the queries were carefully selected to reflect the complexity and retrieval difficulty of real-world usage scenarios. The query types included: (a) parameter lookup (e.g., "What is the design pressure of the reactor coolant system?"), (b) constraint/limit identification (e.g., "What are the allowable concentrations of dissolved oxygen and chloride in reactor coolant water chemistry?"), and (c) multi-hop reasoning across sections (e.g., "How do the design features of the reactor coolant system contribute to safe shutdown under LOCA conditions?").

This pilot-scale evaluation aims to verify the feasibility and effectiveness of the proposed Agentic RAG framework prior to scaling to larger datasets. We also established the conventional Vanilla RAG system as the baseline and conducted performance comparisons against the proposed system.

Table II: Performance Comparison of Agentic RAG vs Vanilla RAG Systems

Metric	Agentic RAG	Agentic RAG	Vanilla RAG	Vanilla RAG
	GPT-4o (API)	Gemma3-27b-IT (Onpremise)	GPT-4o (API)	Gemma3-27b-IT (Onpremise)
Hit@1	0.40	0.50	0.30	0.40
Hit@3	0.70	0.80	0.50	0.60
Precision	0.45	0.52	0.35	0.42
Recall	0.52	0.58	0.38	0.45
F1-Score	0.48	0.55	0.36	0.43

In this study, we employed the Hit@k metric to evaluate retrieval performance, which represents the proportion of queries where relevant documents are included among the top k search results. The experimental results show that our proposed Agentic RAG system achieved overall performance improvements over the conventional Vanilla RAG in both GPT-4o API and Gemma3 on-premises environments. We achieved improvements of 25-33% in Hit@1 and 33-40% in Hit@3, along with consistent performance gains of 24-29% in precision, 29-37% in recall, and 28-33% in F1-Score. Interestingly, the on-premises Gemma-3 environment demonstrated competitive performance compared to the API-based GPT-4o. However, this observation should not be interpreted as inherent superiority of Gemma-3; rather, it is likely influenced by the limited dataset size and the pilot-scale nature of the evaluation, which prevent broad generalization from being drawn. These results suggest that Agentic RAG's multi-step reasoning structure provides certain performance advantages over existing RAG approaches and can serve as a practical alternative in environments where security requirements are critical. Particularly, the findings of this study demonstrate potential applicability to future scenarios requiring extensive technical document analysis and query responses, such as SMART reactor export projects.

4. Conclusions

In this study, we developed and validated an on-premises Agentic RAG system for nuclear safety document analysis. The proposed system can systematically analyze complex FSAR documents through a multi-step reasoning structure of Plan-Execute-Evaluate, demonstrating improved retrieval accuracy and response quality compared to conventional Vanilla RAG systems. Future research will focus on expanding system capacity to handle larger volumes of documents such as SMART nuclear design information and improving the system to accommodate more diverse query types. Additionally, adding multi-modal processing capabilities to enhance technical drawing and graph analysis abilities will be an important development direction.

Acknowledgement

This work was supported by KAERI R&D Program (*KAERI-79755-25*).

REFERENCES

[1] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi et al. Retrieval-Augmented Generation for Large Language Models: A Survey, arXiv preprint arXiv:2312.10997, 2023.

[2] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran at el. ReAct: Synergizing Reasoning and Acting in Language Models, Proceedings of the International Conference on Learning Representations (ICLR), May 1-5, 2023, Kigali, Rwanda.

[3] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal at el. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications, arXiv preprint arXiv:2402.07927, 2024.

[4] G. Team (A. Kamath et al.), Gemma 3 Technical Report, arXiv preprint arXiv:2503.19786, Mar. 2025.

[5] T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia et al. RAFT: Adapting Language Model to Domain Specific RAG, Proceedings of the International Conference on Learning Representations (ICLR), May 7-11, 2024, Vienna, Austria.

[6] P. Verma, S. P. Midigeshi, G. Sinha, A. Solin, N. Natarajan et al. Plan RAG: Efficient Test-Time Planning for Retrieval Augmented Generation, arXiv preprint arXiv:2410.20753, 2024.

[7] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai at el. A Survey on LLM-as-a-Judge, arXiv preprint arXiv:2411.15594, 2024.