Intelligent Predictive Maintenance RAG framework for Power Plants: Enhancing QA with StyleDFS and Domain Specific Instruction Tuning

Seongtae Hong[1*], **Joongmin Shin**[2*], **Jaehyung Seo**[1], **Taemin Lee**[2],
Jeongbae Park[2‡], **Manyoung Cho**[3], **Byeongho Choi**[2], **Heuiseok Lim**[1,2‡]
[1]Department of Computer Science and Engineering, Korea University
[2]Human-inspired AI Research, [3]GaonPlatform
[1,2]{ghdchlwls123,tlswndals13,seojae777,taeminlee,insmile,ilovehci,limhseok}@korea.ac.kr
[3] mycho@gaonpf.com

Abstract

Process plants are complex large-scale industrial facilities that convert raw materials or intermediate products into final products, requiring continuous processes with high safety and efficiency standards. In particular, in nuclear process plants, Predictive Maintenance System (PMS) plays a critical role in predicting equipment anomalies and performing preventive maintenance. However, current PMS relies heavily on the experience of a few experts, leading to knowledge loss upon their retirement and difficulty in swift response. Existing off-premise Question-Answering (QA) systems based on Large Language Models (LLM) face issues such as data leakage and challenges in domain-specific tuning. To address these problems, this study proposes an on-premise intelligent PMS framework utilizing a new chunking method, *StyleDFS*, which effectively reflects the structural information of documents. Additionally, we demonstrate that Instruction tuning using relevant domain-specific data improves LLM performance even under limited data conditions.

1 Introduction

Process plant refers to a large-scale industrial facility that transforms raw materials or intermediate products into finished products through chemical, physical, or biological methods (Jung, 2015; Bajpai, 2018; Miyake et al., 2009). These plants span various industries, including nuclear power plants, each with its unique processes and equipment. Process plant operates on a large scale using continuous or batch processes and requires sophisticated management to enhance safety and efficiency. In nuclear process plants, predictive maintenance systems (PMS) are intelligent technology systems designed to predict and prevent equipment failures by performing preventive maintenance (INGEDULD, 2006; Girotto et al., 2024). These systems analyze operational data and sensor measurements to detect anomalies and plan maintenance activities, thereby improving equipment reliability. Efficient deployment of PMS and prompt execution of appropriate actions ensure the stability of plant operations.

However, the current PMS in nuclear process plants heavily relies on the empirical analysis of a few experts (Gohel et al., 2020; Çınar et al., 2020). The nuclear industry's limited number of experts poses a problem when these experts retire, leading to a loss of valuable knowledge. This reliance on experts negatively impacts the reliability and sustainability of early warning systems. Furthermore, since most tasks in these systems involve repetitive handling of previously occurred issues, expert-dependent methods delay access and analysis of relevant documents, making it difficult to communicate quick responses and adversely affecting the system's overall efficiency.

To address these industrial challenges, existing research has proposed an LLM-based Question-Answering (QA) intelligent system using instruction tuning (Wei et al., 2022; Zhang et al., 2024). Document-based QA systems leveraging large language models (LLMs) have been implemented using off-premise APIs, which reduce dependence on expert knowledge and automate repetitive tasks (Jeong, 2023; Ge et al., 2023; Melz, 2023). However, off-premise solutions include data leakage concerns from the use of external models (Udayakumar and Siddappa, 2010; Chen and Zhao, 2012). Consequently, an on-premise solution utilizing LLM tuning is necessary. Nonetheless, the high-security requirements of the nuclear sector and reliance on expert experience make documenting related knowledge difficult. Additionally, publicly available data is scarce for effective domain-specific tuning (Luo et al., 2023; Jeong, 2023).

In order to resolve the issue of insufficient avail-

---

[*] Equal contributions
[‡] Co-corresponding author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 805–820
November 12-16, 2024 ©2024 Association for Computational Linguistics

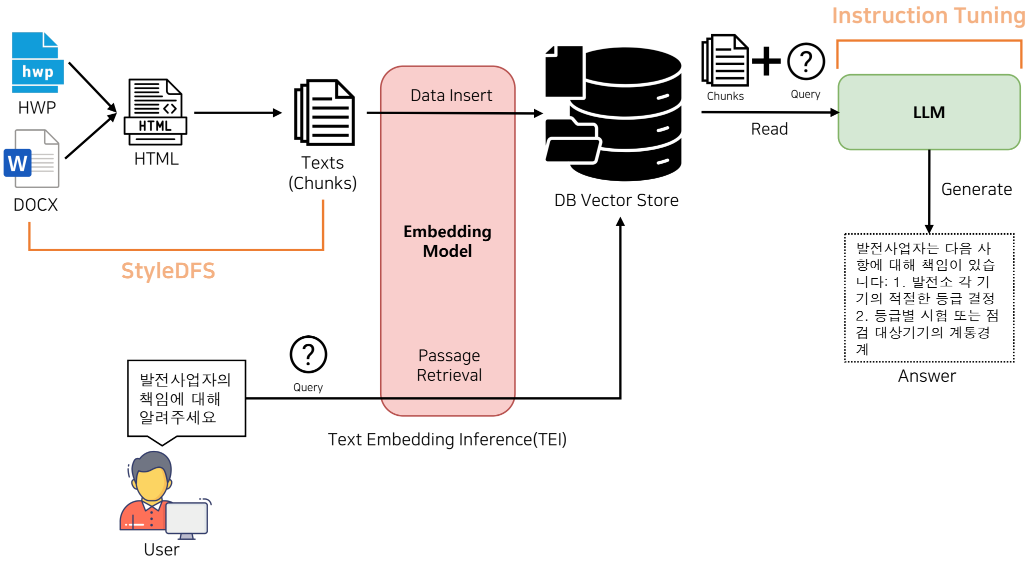

Instruction Tuning
hwp
HWP
W
DOCX
HTML
HTML
Texts
(Chunks)
Data Insert
Embedding
Model
Passage
Retrieval
DB Vector Store
Chunks
Query
Read
LLM
Generate
발전사업자는 다음 사항에 대해 책임이 있습니다: 1. 발전소 각 기기의 적절한 등급 결정 2. 등급별 시험 또는 점검 대상기기의 계통경계
Answer
StyleDFS
발전사업자의 책임에 대해 알려주세요
Query
Text Embedding Inference(TEI)
User

Figure 1: The framework for intelligent PMS using RAG. Documents are converted to HTML and chunked using *StyleDFS*, then stored in a database. The instruction-tuned robust model utilizes the retrieved chunks to generate final answers to user queries. The translated text is as follows: Query: "Please explain the responsibilities of the power plant operator." Answer: "The power plant operator is responsible for the following: 1. Determining appropriate grades for each device in the power plant 2. Performing systematic maintenance for devices based on their grades..."

able data, previous research have utilized Retrieval-Augmented Generation (RAG) (Lewis et al., 2021; Jeong, 2023; Ge et al., 2023; Melz, 2023). However, traditional RAG approaches often assume ideal document formats (e.g., JSON or XML), which is not the case for process plant and corporate or government documents stored as electronic documents (e.g., DOC or HWP) (Kim et al., 2014; K. et al., 2018). Existing chunking methods, such as length and semantic chunking, used when inputting raw electronic documents into a database, fail to adequately consider the structural context of the documents. Moreover, with insufficient domain-specific data, LLMs' comprehension ability is limited, degrading the performance of the PMS.

We propose an intelligent predictive maintenance RAG framework using a new chunking method, *StyleDFS*, which considers the structural information of documents. We enhance LLMs' performance under limited conditions through instruction tuning using publicly available data from relevant scientific and technical domains.

Our main contributions are concluded as follows:

- We propose an automated data processing and efficient retrieval method by a chunking system based on the raw data structure.
- We improve LLMs performance in scenarios with limited domain-specific data by using instruction tuning with relevant domain data.
- We ensure data security and operational reliability by developing an on-premise alert and action framework.

2 Proposed Method

Figure 1 shows the overall structure of the intelligent predictive maintenance RAG framework. This section elaborates on the *StyleDFS* for document chunking and the overall framework.

2.1 StyleDFS for Document Chunking

Electronic documents typically come in formats such as HWP and DOCX, especially in industrial contexts, where these structured formats are prevalent (Lewis et al., 2021; Gao et al., 2024), as shown in Figure 2. Understanding the structural information of documents is crucial for grasping their logical flow and semantic relationships. Existing chunking methods based on length or semantics fail to fully capture the overall structure and elements of documents. To address this issue, we propose the *StyleDFS* chunking method. This approach first converts HWP and DOCX documents into HTML format while preserving their structural information using the Data to HTML conversion module[1]. Subsequently, it performs structure-based chunking

---

[1] `https://github.com/mete0r/pyhwp`

according to Algorithm 1. The algorithm follows these steps:

Tree Structure Transformation Using Style Classes The parsed HTML file separates text using various HTML element tags (e.g., `<div>`, `<p>`, `<span>`). Each tag is automatically assigned a style class in a predefined format by the library, allowing the identification of specific document sections. These style classes enable the transformation of the document into a tree structure. In this tree, each node represents a style class, and branches form connections between parent and child nodes, containing the text for each section. This transformation process clearly defines the hierarchical structure and relationships between elements, preserving structural information. For example, titles, body text, subsections, and lists are distinctly separated within the tree structure. Consequently, the converted HTML tree maintains the overall structure and style of the document, providing a foundation for efficient subsequent traversal.

Chunking with Depth-First Search To traverse the HTML tree and segment the text, we use a preorder Depth First Search (DFS) algorithm. Starting from the root node, DFS visits each node sequentially, accumulating text from leaf nodes and their parents. If the accumulated text exceeds a predefined length (`len`), we add the text up to that point to a chunk list (`chunks`) and start a new chunk. To manage this, we place a flag at the point of exceeding the limit and temporarily pause traversal to begin accumulating a new chunk. The traversal then continues to the right sibling nodes from the current node. The new chunk does not include the contents of the left sibling nodes but retains the parent node's text to maintain consistent context. By recursively visiting child nodes, we continue this accumulation and segment process whenever the text length exceeds the maximum limit. This method ensures consistent maintenance of the hierarchical and semantic context across the document, allowing efficient management of text across multiple sections while preserving structural information.

Algorithm 1 StyleDFS Process: `c_text`: Current text, `p_text`: Parent text, `accum`: Accumulated text

```
Input: html, len
Output: chunks
procedure STYLEDFS(html, len)
    content ← read_file(html)
    tree ← parse_html_to_tree(content)
    chunks ← []
    DFS(tree.root, "", len, chunks, "")
    return chunks
end procedure

function DFS(N, accum, len, chunks, p_text)
    if p_text == "" then c_text ← node.text
    else c_text ← p_text + " " + node.text
    end if
    if accum == "" then accum ← node.text
    else accum ← accum + " " + node.text
    end if
    if length(accum) > len then                         ▷ flag
        append(chunks, accum - node.text)
        accum ← current_text
    end if
    for child in node.children do
        accum ← DFS(N, accum, len, chunks, p_text)
    end for
    if accum ≠ "" and node.isLastChild then
        append(chunks, accum)
        accum ← ""
    end if
    return accum
end function
```

Model	Ko-StrategyQA	Ko-mrtydi	Ko-miracl	Average
multilingual-e5-large	**0.764**	**0.527**	**0.623**	**0.638**
multilingual-e5-base	0.718	0.498	0.585	0.600
multilingual-e5-small	0.698	0.496	0.574	0.589
ko-sroberta-multitask	0.583	0.226	0.297	0.369
UAE-Large-V1	0.061	0.050	0.057	0.056
bge-large-en-v1.5	0.054	0.038	0.047	0.046

Table 1: Performance evaluation results of embedding models on three benchmark tasks using nDCG@3 as the metric.

2.2 RAG Framework

Embedding Model Selection To maximize the performance of the generation model in the RAG, it is essential to effectively retrieve documents highly relevant to the input query. For this purpose, we evaluate retrieval models based on the MTEB leaderboard to select a high-performance Korean embedding model. Table 1 presents the evaluation results, showing that the `multilingual-e5-large` (Wang et al., 2024) outperformed other models, achieving the best performance across three tasks: Ko-StrategyQA, Ko-mrtydi, and Ko-miracl. Detailed information on datasets and metric can be found in Appendix B.

Embedding Storage and Serving Effective implementation of the RAG requires efficient embedding, storage, and rapid retrieval of text data. We extract embedding vectors from chunked documents using the selected model and store them in a PostgreSQL database[2] to support real-time search, and

---

[2] `https://www.postgresql.org/`

additionally, we automate the deployment and serving of the embedding model using the Text Embedding Inference (TEI)[3] toolkit, which includes functionalities such as a web server, load balancer, and worker pool, allowing it to handle multiple users even on a single GPU.

Insturction Tuning In domains like nuclear process plants, where security and safety are critical, a reliable QA system is essential. To generate accurate answers for domain-specific queries, we perform Instruction Tuning using rewritten data pertaining to the science and technology domain. We utilized MRC data from various industrial documents within this domain, converting it into descriptive answers via GPT-4 (OpenAI, 2024). By tuning the model in this way, we enhance the specialized knowledge and expertise within the nuclear, enabling the model to respond accurately to a wide range of potential scenarios. The templates used for Instruction Tuning are presented in Table 6, and detailed explanations of the datasets and conversion tasks are described in §3.

3 Experimental Settings

Dataset To address the need for a QA training dataset in the scientific and technical domain, we utilize the AI-Hub Technical and Scientific Document Reading Comprehension dataset[4]. This dataset comprises short-answer, extractive, and true/false formats. To generate descriptive answers, we use the prompts in Table 6 and employ GPT-4 to rewrite answers for a total of 2,086 samples. The test dataset is composed of the nuclear domain and includes 71 samples with human-created questions, answers, and reference chunks. The documents used for retrieval are approximately 20 pages long and written in HWP format related to the nuclear domain.

Models Our experiments use the pre-trained large language models Llama3-8B (AI@Meta, 2024) and gemma-7B (Team et al., 2024). Additionally, we utilize Llama3-Open-Ko-8B (L, 2024) and gemma-Ko-7B (Junbum Lee, Taekyoon Choi, 2024), which are further pre-trained on a Korean corpus based on the former two models. For retrieving and semantic chunking, we employ multilingual-E5-large (Wang et al., 2024), as discussed in §C

Evaluation Metric To evaluate the model's generated answers to questions using the RAG system, we quantitatively evaluate the quality and accuracy of the answers produced by the models based on each chunking methodology using BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) scores. In our task of generating appropriate responses for PMS, it is essential to capture elements such as contextual appropriateness, fluency, and the ability to generate detailed and extended responses when necessary, which are critical for reliable outputs. To comprehensively evaluate how well the proposed chunking method and other methodologies capture relevant information, we adopt **n-gram based overlap recall** to assess information coverage. This metric indicates how effectively the chunking method organizes relevant information by evaluating how well the retrieved documents include the necessary gold context.

4 Experimental Results

Domain Specific Instruction Tuning Table 2 presents a performance comparison of QA task using various chunking methods. The tuned models exhibit higher performance than the base models in most cases. The models that conduct additional instruction tuning with relevant domain datasets show even greater improvements. This indicates that leveraging a scientific and technical domain dataset to fine-tune the model enhances its performance on the nuclear domain test set.

Effectiveness of Using StyleDFS for RAG System When examining the performance of different chunking methods, semantic-based chunking outperforms length-based chunking in top-1 retrieval scenarios. *StyleDFS* demonstrates the best performance compared to other methods, showing up to a 71% improvement in average category performance. This improvement is based on its ability to segment documents into structurally and contextually meaningful chunks, allowing the retrieval model to find more relevant chunks and the generation model to better utilize them.

As the number of reference chunks increases from top-1 to top-3, there is consistent performance improvement across all methods. Interestingly, in these scenarios, the performance of length-based chunking approaches that of semantic-based chunk-

---

[3] `https://huggingface.co/docs/text-embeddings-inference/index`

[4] `https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=data&dataSetSn=71533`

Methods	Top	Metric	Meta-Llama-3-8B		Llama-3-Open-Ko-8B		gemma-7b		gemma-ko-7b		Average
			base	tune	base	tune	base	tune	base	tune	
Length	1	BLEU	4.080	10.999	2.242	15.203	5.905	10.689	1.792	13.796	8.088
		ROUGE-L	8.433	6.700	3.332	12.065	6.118	7.886	2.177	10.370	7.259
	3	BLEU	5.665	16.235	3.598	18.973	4.942	10.330	3.263	20.449	10.556
		ROUGE-L	*9.345*	12.417	*7.015*	*17.966*	7.669	6.393	5.041	14.124	9.746
Semantic	1	BLEU	**6.219**	11.704	3.038	16.993	4.005	10.077	1.409	15.239	8.085
		ROUGE-L	7.551	7.305	4.356	13.710	7.841	7.672	2.074	8.821	7.041
	3	BLEU	5.913	17.703	**4.425**	20.646	4.075	10.954	3.000	19.976	10.961
		ROUGE-L	8.049	9.423	5.940	13.806	7.587	9.694	4.026	13.245	8.971
StyleDFS	1	BLEU	3.356	13.671	3.120	19.072	5.924	16.514	5.397	17.510	10.070
		ROUGE-L	6.235	13.746	6.012	13.046	*11.005*	14.773	7.102	13.818	10.216
	3	BLEU	3.479	**19.579**	2.705	**23.138**	**6.626**	**17.595**	**8.401**	**20.486**	**12.500**
		ROUGE-L	5.098	*18.245*	5.405	16.949	8.547	*13.084*	*9.242*	*16.447*	*11.377*
Gold	-	BLEU	10.049	24.103	4.267	31.612	11.798	21.479	8.030	29.486	17.478
		ROUGE-L	9.939	21.200	7.947	22.930	16.409	13.676	10.027	15.436	14.945

Table 2: This table compares performance across various models and methods, measured by BLEU and ROUGE-L metrics. It presents results for both base and tuned configurations of four models. The analysis categorizes performance into segments such as Length, Semantic, and *StyleDFS*, offering clear benchmarks for comparison in both Top 1 and Top 3 retrieval settings. The "Gold" section displays the generation results when provided with chunks containing the correct answers. In each column, the highest BLEU score is is indicated by **bold** and the ROUGE-L score by *italics*.

ing. This result shows that the reference chunks include more tokens, providing the generation model with sufficient context. However, our proposed method still outperforms others significantly. Even with more reference documents, the top-3 performance of length and semantic-based chunking does not surpass the top-1 performance of *StyleDFS*. This indicates that our method segments documents not merely by individual elements (tags) but by considering the entire document structure and context. These findings emphasize that the generation performance of QA systems in RAG framework depends on accurate and context-rich input chunks. *StyleDFS* effectively divides these chunks to achieve performance close to that of gold reference chunks, highlighting its superior capability in structuring and contextualizing documents.

Methods	Top	1-gram	2-gram	3-gram
Length	1	0.2023	0.1358	0.1066
	3	0.3668	0.2404	0.1862
Semantic	1	0.2464	0.1642	0.1281
	3	0.4363	0.2935	0.2294
StyleDFS	1	**0.3595**	0.2975	0.2727
	3	**0.5710**	0.4716	0.4326

Table 3: Recall rates for various chunking methods across different n-gram lengths, presented at top-1 and top-3 settings. **Bold** indicate the highest performance achieved by any method for the respective Top setting

Information Coverage Analysis Table 3 presents the information coverage for the chunking methods, measured by how well the retrieved documents from test set queries capture the context relevant to the correct chunks. A higher overlap recall score indicates that the retrieved document contains more relevant context compared to the original document. Across all n-gram and top-k settings, the semantic-based chunking method exceeds the length-based method. Notably, the *StyleDFS* outperforms all other chunking methods in the same settings. For instance, in the 1-gram chunks, it achieves a recall of 0.3595 in the top-1 setting and 0.5710 in the top-3 setting, significantly higher than other methods. Similarly, the results demonstrate higher coverage in the 2-gram and 3-gram chunks. This consistent superiority across various n-gram settings highlights the robustness and reliability of our proposed method in different scenarios. Moreover, as shown in Table 4, although our approach segments the highest number of chunks, the retrieval model skillfully selects the relevant chunks. These results suggest that our method more effectively includes the correct chunks within the extracted chunks.

4.1 Ablation Study

To clarify the individual contributions of the proposed *StyleDFS* chunking method and instruction tuning, we conducted ablation studies focusing on these components. As shown in Table 2, *StyleDFS* consistently outperformed both length-based and semantic-based chunking methods when using the base models without instruction tuning. This significant improvement demonstrates that our proposed method is capable of producing chunks that are more contextually relevant and structurally consistent, both of which are crucial elements for effective information retrieval and generation within RAG systems. Additionally, *StyleDFS* maintained its superiority even when the number of reference chunks was increased, achieving the highest scores without the need to increase the top-k settings. This indicates that the chunks generated by *StyleDFS* are highly relevant, allowing the retrieval model to effectively exploit them and reducing the necessity to retrieve multiple chunks for optimal performance.

Furthermore, we evaluated the impact of instruction tuning by comparing tuned models with their respective base counterparts. Instruction tuning led to significant performance improvements across all chunking methods, highlighting its crucial role in adapting the model to specific linguistic and domain-related nuances. This enhancement enables the model to generate more accurate and contextually appropriate responses. Moreover, when instruction tuning was combined with the *StyleDFS* chunking method, its effects were further amplified, resulting in the highest performance metrics among all tested configurations. This synergy between effective chunking and tailored instruction tuning emphasizes the importance of both components in building robust and high-performing QA systems within the RAG framework.

5 Related Work

Predictive Maintainance System PMS is essential in various industries, particularly in high-risk sectors like nuclear process plants, where they play a crucial role in ensuring reliability and safety. Traditional research has primarily relied on expert empirical analysis and conventional data analysis methods, which come with limitations such as the scarcity of experts and the risk of information loss. Gohel et al. (2020) and Çınar et al. (2020) warn that the retirement or absence of experts can degrade the system's reliability and efficiency. Recent studies have focused on developing intelligent systems using LLMs. LLM-based QA systems can leverage large datasets to model expert knowledge and automate problem-solving. Research by Jeong (2023), Ge et al. (2023), and Melz (2023) demonstrates that document-based QA systems using LLMs are effective in reducing dependency on experts and automating repetitive tasks. However, these systems, implemented as off-premise solutions using external models, pose data leakage risks (Udayakumar and Siddappa, 2010; Chen and Zhao, 2012). To address this, on-premise solutions have been proposed, but securing sufficient domain-specific training data remains a significant challenge. Lewis et al. (2021) and Gao et al. (2024) have shown that the RAG approach, which combines document retrieval and generation processes, can enhance LLM performance. Nonetheless, the high-security requirements in the nuclear field and the difficulty in documenting expert knowledge pose challenges for adoption (Luo et al., 2023; Jeong, 2023). Additionally, there are limitations in handling the established document formats used in actual industrial settings (Kim et al., 2014; K. et al., 2018).

Chunking Method The performance of retrieval models is influenced by the chunking strategy used to segment documents (Duarte et al., 2024). Length-based chunking, which divides documents into fixed lengths, is simple and fast to implement but often disregards the document's inherent structure, disrupting its logical flow. This method does not consider the semantic connections between data, which can cause problems in practical applications (Gong et al., 2020). Other approaches involve splitting documents based on specific criteria such as line breaks, spaces, or punctuation (Langchain, 2023). Alternatively, semantic-based chunking uses encoder models like BERT (Devlin et al., 2018) to segment documents based on their meaning. While this method can capture semantic information, it is dependent on the model's performance and can be time-consuming and costly. It identifies split points based on semantic understanding but still struggles to fully reflect the document's overall structure (Devarajan and Subramanian, 2022). Recently, a new method called LumberChunker (Duarte et al., 2024) has been proposed, which directly uses LLMs to determine dynamic segmentation points in a document, focusing on maintaining semantic coherence. This method aims to retain the semantic consistency of the document but also falls short in perfectly capturing both semantic and structural information. Traditional chunking methods often result in a loss of context and a decrease in information consistency due to their inability to fully integrate the document's semantic and structural elements.

6 Conclusion

In this paper, we propose a practical intelligent PMS to address the major issues present in existing systems for process plants. Traditional systems heavily rely on experts, leading to reliability and sustainability problems. The manual processes involve also reduce responsiveness and efficiency. Unstructured and complex document formats in the nuclear industry make it difficult to use existing chunking and retrieval methods effectively, revealing limitations in the current RAG frameworks. We introduce the *StyleDFS*, which converts documents into a structured format and parses them efficiently. This approach segments documents into structurally and semantically related chunks, significantly improving information recall and generation performance. In the future work, we will focus on applying this framework across various industries to validate its performance and continually enhance the capabilities of state-of-the-art LLM models to increase the effectiveness of intelligent predictive maintenance systems.

Limitations

This study presents several limitations. First, securing datasets for the nuclear industry is exceedingly difficult due to its closed nature, restricted access, and stringent security requirements. The limited number of experts with clearance to handle the data significantly hinders both the collection of sufficient data necessary for optimizing model performance and the human validation of the generated results. Second, our experiment was constrained by the typical computing resource limitations of an on-premises execution environment, particularly the use of a single GPU. Consequently, in the current study, we focused on models with backbones smaller than 10B parameters. Third, we did not extensively address various document formats. Many documents are digitized in non-standard formats, complicating the processing. Additionally, these documents contain numerous specialized terminologies, further increasing reliance on experts. These limitations may restrict the generalizability of the research findings. Future research should focus on securing more comprehensive datasets and validating the approach across various document formats to overcome these challenges and enhance the applicability of the study's results.

Ethical considerations

Our research addresses the development of an intelligent predictive maintenance system for the nuclear domain, prioritizing security. Due to the nature of the nuclear domain, we cannot directly use the entire dataset for training. Only a subset of data samples and documents is made available within the permitted scope for evaluation. The study adheres to relevant laws and data protection standards rigorously. LLMs underwent extensive validation to ensure reliability and safety. We particularly emphasize the importance of on-premise environments, and this research aims to enhance data security and operational reliability.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT). (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI) This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425).This paper was also carried out as a research project under the contract research center for SaaS LLM (Software-as-a-Service based on Large Language Models) by Gaon Platform and Korea University.

References

AI@Meta. 2024. Llama 3 model card.

P. Bajpai. 2018. *Biermann's Handbook of Pulp and Paper: Volume 1: Raw Material and Pulp Making*. Elsevier Science.

Deyan Chen and Hong Zhao. 2012. Data security and privacy protection issues in cloud computing. In *2012 International Conference on Computer Science and Electronics Engineering*, volume 1, pages 647–651.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Viji Devarajan and R. Subramanian. 2022. Analyzing semantic similarity amongst textual documents to suggest near duplicates. *Indonesian Journal of Electrical Engineering and Computer Science*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

André V Duarte, João Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L Oliveira. 2024. Lumberchunker: Long-form narrative document segmentation. *arXiv preprint arXiv:2406.17526*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

J. Ge, Steve Sun, Joseph Owens, Victor Galvez, O. Gologorskaya, Jennifer C Lai, Mark J Pletcher, and Ki Lai. 2023. Development of a liver disease-specific large language model chat interface using retrieval augmented generation. *medRxiv*.

Cristiane D. Girotto, Farzad Piadeh, Vahid Bkhtiari, Kourosh Behzadian, Albert S. Chen, Luiza C. Campos, and Massoud Zolgharni. 2024. A critical review of digital technology innovations for early warning of water-related disease outbreaks associated with climatic hazards. *International Journal of Disaster Risk Reduction*, 100:104151.

Hardik A. Gohel, Himanshu Upadhyay, Leonel Lagos, Kevin Cooper, and Andrew Sanzetenea. 2020. Predictive maintenance architecture development for nuclear infrastructure using machine learning. *Nuclear Engineering and Technology*, 52(7):1436–1442.

Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent chunking mechanisms for long-text machine reading comprehension. pages 6751–6761.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020. Kornli and korsts: New benchmark datasets for korean natural language understanding. *arXiv preprint arXiv:2004.03289*.

PETR INGEDULD. 2006. Real time analysis for early warning systems. In *Security of Water Supply Systems: from Source to Tap*, pages 65–84, Dordrecht. Springer Netherlands.

CheonSu Jeong. 2023. A study on the implementation of generative ai services using an enterprise data-based llm application architecture. *Adv. Artif. Intell. Mach. Learn.*, 3:1588–1618.

Junbum Lee, Taekyoon Choi. 2024. gemma-ko-7b.

Seungho Jung. 2015. Facility siting and plant layout optimization for chemical process safety. *Korean Journal of Chemical Engineering*, 33:1–7.

Rajbabu K., Harshavardhan Srinivas, and Sudha S. 2018. Industrial information extraction through multi-phase classification using ontology for unstructured documents. *Computers in Industry*, 100:137–147.

Chang-Su Kim, Kyu-Chul Shim, Byoung-Jun Kang, Kyung-Hwan Kim, and Hoe-Kyung Jung. 2014. Design and implementation of input and output system for unstructured big data. *Journal of the Korea Institute of Information and Communication Engineering*, 18.

Junbum L. 2024. Llama-3-open-ko.

Langchain. 2023. Recursive character text splitter documentation.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Augmented large language models with parametric knowledge guiding. *arXiv preprint arXiv:2305.04757*.

Eric Melz. 2023. Enhancing llm intelligence with arm-rag: Auxiliary rationale memory for retrieval augmented generation. *ArXiv*, abs/2311.04177.

Chikahiro Miyake, Katsumi Amako, Naomasa Shiraishi, and Toshio Sugimoto. 2009. Acclimation of Tobacco Leaves to High Light Intensity Drives the Plastoquinone Oxidation System—Relationship Among the Fraction of Open PSII Centers, Non-Photochemical Quenching of Chl Fluorescence and the Maximum Quantum Yield of PSII in the Dark. *Plant and Cell Physiology*, 50(4):730–743.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Taehwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Lucy Park, Alice Oh, Jungwoo Ha, and Kyunghyun Cho. 2021. Klue: Korean language understanding evaluation. *Preprint*, arXiv:2105.09680.

Yingxia Shao Shitao Xiao, Zheng Liu and Zhao Cao. 2022. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. In *EMNLP*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

NL Udayakumar and M Siddappa. 2010. Security issues and solutions for virtualization in cloud computing service. *In International Journal for Engineering Research & Technology (IJERT), 2010*, pages 55–57.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pages 25–54. PMLR.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction tuning for large language models: A survey.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. tydi: A multi-lingual benchmark for dense retrieval. *arXiv preprint arXiv:2108.08787*.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

Zeki Murat Çınar, Abubakar Abdussalam Nuhu, Qasim Zeeshan, Orhan Korhan, Mohammed Asmael, and Babak Safaei. 2020. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, 12(19).

A Chunking Results Statistics

Statistic Type		**Length**	**Semantic**	**StyleDFS**
Character	Total	30,493	30,493	30,483
	Avg	417.71	586.40	247.82
	Std	18.64	128.33	170
	Mid	419	624	184
	Max	443	670	679
	Min	278	32	47
Token	Total	21,290	21,259	20,839
	Avg	291.64	408.82	169.422
	Std	20.01	91.93	118.81
	Mid	289	433	126
	Max	366	490	449
	Min	181	23	28
Chunk	Total	73	52	123

Table 4: Statistical results for the three chunking methods (Length, Semantic, and *StyleDFS*) categorized by Character, Token, and Chunk.

Table 4 shows the statistical results for the three chunking methods: Length, Semantic, and *StyleDFS*. Length-based chunking generates a total of 73 chunks, with an average of 417.71 characters and 291.64 tokens per chunk, showing a uniform tendency. The low standard deviations of 18.64 for characters and 20.01 for tokens indicate relatively little variability in chunk length, which results from splitting the text based on a fixed length.

In contrast, semantic-based chunking, using an encoder model, produces a total of 52 chunks. This method groups chunks based on the semantic similarity of the text, resulting in the highest average character count of 586.40 and average token count of 408.82. However, the standard deviations of 128.33 for characters and 91.93 for tokens indicate greater variability in chunk length, showing the larger sections based on semantic similarity.

Lastly, our proposed *StyleDFS* generates a total of 123 chunks, the highest number among the methods. It has the lowest average character count of 247.82 and token count of 169.42, with standard deviations of 170 and 118.81, respectively, indicating significant variability in chunk length. The large difference between the maximum length (679 characters) and minimum length (47 characters) shows that *StyleDFS* finely captures the hierarchical structural elements of HTML documents in its chunking process.

B Embedding Model Selection Details

MTEB (Muennighoff et al., 2022) is a comprehensive text embedding benchmark designed to evaluate embedding models. However, since it does not support retrieval evaluation in Korean, we extended MTEB by including three Korean retriever performance evaluation datasets to assess the performance of Korean embedding models.

Datasets To evaluate the embedding models, we used three datasets: Ko-StrategyQA[5], Ko-mrtydi[6], and Ko-miracl[7]. To create these datasets, we first converted the Korean (Ko) sections of the Mr.TyDI (Zhang et al., 2021) and Miracl (Zhang et al., 2023) datasets into the BeIR (Thakur et al., 2021) format. The Ko-StrategyQA dataset, designed for multi-hop QA, was also converted into the BeIR format. This process included grouping evidence documents and excluding sub-questions containing no_evidence or operation to refine the dataset further.

Models The multilingual-e5 (Wang et al., 2024), initialized from xlm-roberta (Conneau et al., 2019), was continually trained on a mixture of multilingual datasets, supporting 100 languages originally covered by xlm-roberta and featuring a hidden size of 1024. This model's broad language support makes it highly versatile. ko-sroberta[8], a sentence-transformers model that maps sentences and paragraphs to a 768-dimensional dense vector space. This model extends klue/roberta-base (Park et al., 2021) through multi-task learning using the Ko-rSTS and KorNLI datasets in Ham et al. (2020), making it particularly effective for tasks involving Korean semantic representation. Additionally, we leveraged the UAE-Large-V1 (Li and Li, 2023), which is based on the BERT architecture and has a hidden size of 1024. This model introduces the AnglE method, a novel angle-optimized text embedding approach that mitigates the adverse effects of the cosine function's saturation zone by applying angle optimization in a complex space, thereby enhancing the quality of its semantic embeddings. Furthermore, we included the bge-large-en-v1.5 (Xiao et al., 2023) in our experiments. This model also follows the BERT architecture with a hidden size of 1024 and is pre-trained using retromae (Shitao Xiao and Cao, 2022). It undergoes further training on large-scale paired data through contrastive learning, refining its ability to generate high-quality embeddings for retrieval and matching tasks.

Metrics To select and evaluate the retrieval models, we use the nDCG (Normalized Discounted Cumulative Gain) (Wang et al., 2013) score. The nDCG metric assesses the relevance of retrieved documents by considering their rank, measuring the quality and order of the search results. Specifically, we focus on the top 3 retrieved documents to determine how effectively the model ranks the most relevant documents for a query. This approach provides a detailed evaluation of the precision and ranking efficiency of the retrieval models, offering the necessary metric to choose the most suitable model for our application.

Latency We evaluate the encoding throughput using the Ko-mrtydi dataset (1,496,126 paragraphs). The time required for this task using the Multilingual-e5 encoding model is detailed below.

Model	Time (sec)	Throughput (it/s)
Multilingual-e5-small	1093.18	1368
Multilingual-e5-base	2564.65	583
Multilingual-e5-large	8069.48	185

Table 5: Evaluation of encoding throughput for the top 3 high-performing models in 1 using the Ko-mrtydi dataset. The table presents the total time taken to encode all samples (in seconds) and the throughput (iterations per second).

Hardware We conducted our experiments using an Intel Xeon Gold 6230R @2.10GHz CPU, 376GB RAM, and an NVIDIA RTX A6000 48GB GPU. The software environment included nvidia-driver, CUDA, and PyTorch, running on Ubuntu 20.04.6 LTS.

C Experimental Details

Models We utilized several state-of-the-art language models in our experiments. The LLaMA-8B model, comprises 8 billion parameters and is trained on over 15 trillion tokens of data from publicly available sources, incorporating non-English data in approximately 30 languages. Additionally,

---

[5] https://huggingface.co/datasets/taeminlee/Ko-StrategyQA

[6] https://huggingface.co/datasets/taeminlee/Ko-mrtydi

[7] https://huggingface.co/datasets/taeminlee/Ko-miracl

[8] https://huggingface.co/jhgan/ko-sroberta-multitask

we employed the LLaMA Open Ko 8, a Korean-adapted version of LLaMA-3-8B, which continued training to over 60GB of publicly available, deduplicated texts. The Gemma 7B model, trained on a diverse dataset of 6 trillion tokens encompassing web documents, code, and mathematical texts, offers a broad exposure to various linguistic styles and vocabulary. Furthermore, we included the Gemma-Ko 7B, a Korean-language adaptation of the Gemma 7B model.

Hyperparameters The model is trained using the Adam optimizer with a learning rate (LR) of 2e-5. The learning rate scheduler employed is WarmupDecayLR. Maximum sequence length for the model is set to 2048 tokens, and bfloat16 precision is used for the computations. The training utilizes a batch size of 16 and is conducted over 3 epochs. Additionally, a warmup phase is implemented for the initial 10% of the training steps.

Hardware We utilized 8 NVIDIA A6000 GPU with 48GB memory capacity and AMD EPYC 7513 32-core Processor CPUs to training the LLMs. For inference we use a singlie accelerator.

D Prompt Template

Answer rewriting prompt Table 6 provides an example of the prompt template used for answer rewriting. This template is designed to elicit detailed and descriptive-form responses, converting the original yes/no and extractive formats into more comprehensive and descriptive answers.

You need to create a set of Q&As that train linguistic intelligence.

It includes four basic skills
C=given context
Q=given questions
A=create answers
J=formatting to json

The context must meet the following requirements
Question creation must meet the following requirements
1. Questions are built with a good understanding of context.
2. Don't be edited.

Answer generation must meet the following requirements
1. provide a rationale.
2. be formatted in markdown.
3. Must be written in Korean and descriptive form.

Formatting to json must meet the following requirements
1. have three keys: context, question and answer.

Perform tasks C, Q, A, and J for the following documents.
Imporant: always use the response tool to respond to the user.

Context: 수중 통신망의 매체접속제어 (MAC: Medium Access Control) 프로토콜 설계 시 반드시 고려되어야할 사항은 초음파의 느린 속도로 인한 긴 전파 지연, 동기화의 어려움, 그리고 수중 환경에서 배터리 충전의 어려움으로 인한 전력 소비 문제 등이다. 본 논문에서는 TDMA 기반의 매체접속 제어 프로토콜이 가지는 동기화 문제, 채널 효율 문제를 해결하고, 경쟁기반 프로토콜이 가지는 충돌율로 인한 전송 효율 저하 문제를 보완하는 매체접속 제어 프로토콜을 제안하였다. 슬립 모드를 도입하여 효율적인 전력 사용으로 에너지 소비를 줄였고, 채널 효율 증가 및 충돌율 감소를 통해 전송 효율을 증가시켰다.

Question: 이 논문에서 제안한 프로토콜은 어떤 방법의 매체접속 제어 프로토콜이야?

Table 6: Example of a prompt template used for rewriting answers in the dataset. Text highlighted in red represents instructions, yellow represents the context, and green represents the question.

QA prompt In the RAG pipeline, the input to the LLM consists of context chunks followed by the query. For the top-1 scenario, a single chunk is used, whereas for the top-3 scenario, chunks are concatenated in order of their relevance with newline separators. An actual example can be seen in Table 7, which illustrates the format utilized for the LLMś input.

E Examples

Results of Document Chunking Using Different Methods Table 8 displays the results of applying each chunking methodology to the document shown in Figure 2. The length based approach divides the word '발전소' into separate chunks. The semantic based method, while considering meaning, fails to form a complete chunk in the given example. This issue arises because the method concatenates chunks based on the generation probability of the model in a one-way manner. In contrast, our methodology successfully generates a document that is structurally and semantically complete.

Comparison of Generation Results Table 9 shows the results generated by augmenting top-1 documents chunk using different methods. Length-based chunking tends to truncate important parts of documents, omitting crucial information needed to answer queries. This leads to incomplete or inconsistent responses. For instance, queries about "the responsibilities of the power project operator" resulted in partial or abruptly cut-off information. Semantic-based chunking considers the meaning of the document but still struggles to form complete chunks. Because the chunking is linked based on the model's generation probability, even though important information is included, it fails to function

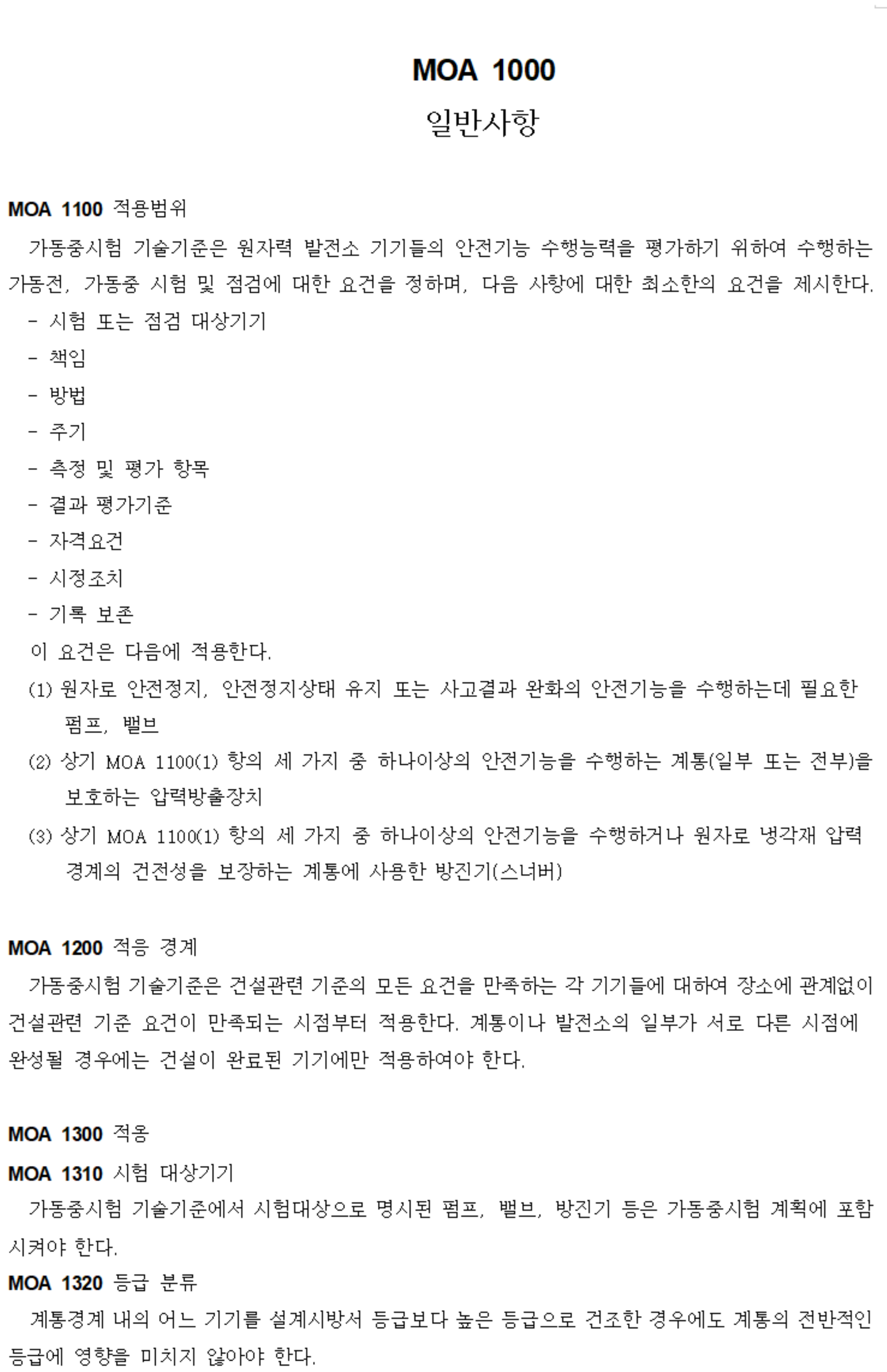
```
## MOA 3000 일반요건\n ### MOA 3120 가동중점검 및 시험기간\ n (1) 가동중점검 및 시험주기는 가동중시험 기술기준의 요건에 따라야 한다. (2) 가동중점검 및 시험기간은 상업운전 이후의 햇수로 결정하여야 한다.
...
대안으로 가동전시험 계획은 규제기관에서 정한 후속 기준판 및 추록을 만족하여야 한다. 이 후속 기준판 및 추록의 특정부분 만을 적용할 수 도 있으나 , 이를 위하여서는 특정부분의 관련요건을 모두 만족하여야 한다.

가동전시험 기간이 뭐야?
```

Table 7: Example of a prompt commonly used for both training and inference in the RAG pipeline. The text above the newline represents the chunk(s) and below represents the question.

as a fully independent document. While this approach adequately answers queries, some responses lack consistency. *StyleDFS* effectively reflects both the structural and semantic context of documents, generating more complete and coherent chunks. For example, in response to questions about "the responsibilities of the power project operator," it organizes and presents information from multiple parts of the document in a clear and structured manner. This method significantly improves the quality and consistency of responses by efficiently incorporating overall document content. The document retrieval is accurate, and the necessary information is well-captured, leading to successful generation. Table 10 provides the English translations.

Length-based	**Chunk 1**	MOA 1000 일반사항 MOA 1100 적용범위 가동중시험 기술기준은 원자력 발전소 기기들의 안전기능 수행능력을 평가하기 위하여 수행하는 가동전, 가동중 시험 및 점검에 대한 요건을 정하며, 다음 사항에 대한 최소한의 요건을 제시한다. - 시험 또는 점검 대상 기기 - 책임 - 방법 - 주기 - 측정 및 평가 항목 - 결과 평가기준 - 자격요건 - 시정조치 - 기록 보존 요건은 다음에 적용한다. (1) 원자로 안전정지, 안전정지상태 유지 또는 사고결과 완화의 안전기능을 수행하는데 필요한 펌프, 밸브 (2) 상기 MOA 1100 (1) 항의 세 가지 중 하나 이상의 안전기능을 수행하는 계통(일부 또는 전부)을 보호하는 압력방출장치 (3) 상기 MOA 1100 (1) 항의 세 가지 중 하나 이상의 안전기능을 수행하거나 원자로 냉각재 압력경계의 건전성을 보장하는 계통에 사용한 방진기(스너버) MOA 1200 적용 경계 가동중시험 기술기준은 건설관련 기준의 모든 요건을 만족하는 각 기기들에 대하여 장소에 관계없이 건설관련 기준 요건이 만족되는 시점부터 적용한다. 계통이나 발전
	Chunk 2	소의 일부가 서로 다른 시점에 완성될 경우에는 건설이 완료된 기기에만 적용하여야 한다. MOA 1300 적용 MOA 1310 시험 대상기기 가동중시험 기술기준에서 시험대상으로 명시된 펌프, 밸브, 방진기 등은 가동중시험 계획에 포함시켜야 한다. MOA 1320 등급 분류 계통경계 내의 어느 기기를 설계시방서 등급보다 높은 등급으로 건조한 경우에도 계통의 전반적인 등급에 영향을 미치지 않아야 한다. 참조기준 또는 규격 개정일자 / 년도 PTC 25 API RP-527 1994년 3판, 1991년 MOA 1400 참조 기준 및 규격 가동중시험 기술기준의 참조 기준과 규격의 개정일자 및 발행년도는 표 MOA 1400 과 같다. 표 MOA 1400 참조 기준 및 규격 MOA 1500 발전사업자의 책임 발전사업자는 다음 사항에 대하여 책임이 있다. (1) 발전소 각 기기의 적절한 등급 결정 (2) 등급별 시험 또는 점검 대상기기의 계통경계 선정 (3) 시험 및 점검 면제 대상기기 선정 (4) 시험 및 점검 수행에 적절한 접근로와 공간을 확보하도록 기기를 설계 및 배치 부록 B 임
Semantic-based	**Chunk 1**	MOA 1000 일반사항 MOA 1100 적용범위 가동중시험 기술기준은 원자력 발전소 기기들의 안전기능 수행능력을 평가하기 위하여 수행하는 가동전, 가동중 시험 및 점검에 대한 요건을 정하며, 다음 사항에 대한 최소한의 요건을 제시한다. - 시험 또는 점검 대상기기 - 책임 - 방법 - 주기 - 측정 및 평가 항목 - 결과 평가기준 - 자격요건 - 시정조치 - 기록 보존 요건은 다음에 적용한다. (1) 원자로 안전정지, 안전정지상태 유지 또는 사고결과 완화의 안전기능을 수행하는데 필요한 펌프, 밸브 (2) 상기 MOA 1100 (1) 항의 세 가지 중 하나 이상의 안전기능을 수행하는 계통(일부 또는 전부)을 보호하는 압력방출장치 (3) 상기 MOA 1100 (1) 항의 세 가지 중 하나 이상의 안전기능을 수행하거나 원자로 냉각재 압력경계의 건전성을 보장하는 계통에 사용한 방진기(스너버) MOA 1200 적용 경계 가동중시험 기술기준은 건설관련 기준의 모든 요건을 만족하는 각 기기들에 대하여 장소에 관계없이 건설관련 기준 요건이 만족되는 시점부터 적용한다. 계통이나 발전소의 일부가 서로 다른 시점에 완성될 경우에는 건설이 완료된 기기에만 적용하여야 한다. MOA 1300 적용 MOA 1310 시험 대상기기 가동중시험 기술기준에서 시험대상으로 명시된 펌프, 밸브, 방진기 등은 가동중시험 계획에 포함시켜야 한다. MOA 1320 등급 분류 계통경계 내의 어느 기기를 설계시방서 등급보다 높은 등급으로 건조한 경우에도 계통의 전반적인 등급에 영향을 미치지 않아야 한다. 참조기준 또는 규격 개정일자 / 년도 PTC 25 API
	Chunk 2	RP-527 1994년 3판, 1991년 MOA 1400 참조 기준 및 규격 가동중시험 기술기준의 참조 기준과 규격의 개정일자 및 발행년도는 표 MOA 1400 과 같다. 표 MOA 1400 참조 기준 및 규격 MOA 1500 발전사업자의 책임 발전사업자는 다음 사항에 대하여 책임이 있다. (1) 발전소 각 기기의 적절한 등급 결정 (2) 등급별 시험 또는 점검 대상기기의 계통경계 선정 (3) 시험 및 점검 면제 대상기기 선정 (4) 시험 및 점검 수행에 적절한 접근로와 공간을 확보하도록 기기를 설계 및 배치 부록 B 임의요건을 지침으로 참조 (5) 시험 계획 및 (6) 시험 및 점검일정의 수립 지침 및 절차서 작성 (7) 발전사업자의 품질보증계획에 따라 점검 및 시험을 수행, 평가하는 자의 자격 인정 (8) 시험 및 점검 수행 (9) 평가 근거를 마련하고, 향후 시험 또는 점검 결과와 비교가 가능하도록 결과를 기록 (10) 시험 및 점검 결과 평가 (11) 적절한 시험 및 점검 기록(시험 및 점검 결과 및 절차 기술) 유지 (12) 기기 또는 계통 수명기간 동안 시험 및 점검 결과 보존 (13) KEPIC-QAP에 따른 품질보증계획 문서화 MOA 1600 접근성 시험 및 점검 수행에 필요한 점검자 및 장비의 접근성이 확보되어야 한다. 주(1) 등급분류는 원자력안전위원회 고시 제2014-15(원자로시설의 안전등급과 등급별 규격에 관한 규정)에 따른다. MOA 2000 용어정의 가동전시험(PST) : 기기 설치 후부터 노심열에 의한 최초 전력생산 이전까지의 기간 또는 가동중 원전에서 기기 가동전 기간에 수행하는 시험 가동전시험 기간 : 기기 설치
StyleDFS	**Chunk 1**	## MOA 1000 일반사항 ### MOA 1100 적용범위 가동중시험 기술기준은 원자력 발전소 기기들의 안전기능 수행능력을 평가하기 위하여 수행하는 가동전, 가동중 시험 및 점검에 대한 요건을 정하며, 다음 사항에 대한 최소한의 요건을 제시한다. - 시험 또는 점검 대상기기 - 책임 - 방법 - 주기 - 측정 및 평가 항목 - 결과 평가기준 - 자격요건 - 시정조치 - 기록 보존 요건은 다음에 적용한다. (1) 원자로 안전정지 안전정지상태 유지 또는 사고결과 완화의 안전기능을 수행하는데 필요한 펌프 밸브 (2) 상기 MOA 1100 (1) 항의 세 가지 중 하나 이상의 안전기능을 수행하는 계통(일부 또는 전부)을 보호하는 압력방출장치 (3) 상기 MOA 1100 (1) 항의 세 가지 중 하나 이상의 안전기능을 수행하거나 원자로 냉각재 압력경계의 건전성을 보장하는 계통에 사용한 방진기(스너버)
	Chunk 2	## MOA 1000 일반사항 ### MOA 1200 적용 경계 가동중시험 기술기준은 건설관련 기준의 모든 요건을 만족하는 각 기기들에 대하여 장소에 관계없이 건설관련 기준 요건이 만족되는 시점부터 적용한다. 계통이나 발전소의 일부가 서로 다른 시점에 완성될 경우에는 건설이 완료된 기기에만 적용하여야 한다.

Table 8: Qualitative comparison of chunking methods applied to the document in Figure 2. The table allows a direct observation of the chunking results for each methodology: length-based, semantic-based, and *StyleDFS*.



MOA 1000

일반사항

MOA 1100 적용범위

가동중시험 기술기준은 원자력 발전소 기기들의 안전기능 수행능력을 평가하기 위하여 수행하는 가동전, 가동중 시험 및 점검에 대한 요건을 정하며, 다음 사항에 대한 최소한의 요건을 제시한다.

- 시험 또는 점검 대상기기
- 책임
- 방법
- 주기
- 측정 및 평가 항목
- 결과 평가기준
- 자격요건
- 시정조치
- 기록 보존

이 요건은 다음에 적용한다.

(1) 원자로 안전정지, 안전정지상태 유지 또는 사고결과 완화의 안전기능을 수행하는데 필요한 펌프, 밸브

(2) 상기 MOA 1100(1) 항의 세 가지 중 하나이상의 안전기능을 수행하는 계통(일부 또는 전부)을 보호하는 압력방출장치

(3) 상기 MOA 1100(1) 항의 세 가지 중 하나이상의 안전기능을 수행하거나 원자로 냉각재 압력 경계의 건전성을 보장하는 계통에 사용한 방진기(스너버)

MOA 1200 적응 경계

가동중시험 기술기준은 건설관련 기준의 모든 요건을 만족하는 각 기기들에 대하여 장소에 관계없이 건설관련 기준 요건이 만족되는 시점부터 적용한다. 계통이나 발전소의 일부가 서로 다른 시점에 완성될 경우에는 건설이 완료된 기기에만 적용하여야 한다.

MOA 1300 적용

MOA 1310 시험 대상기기

가동중시험 기술기준에서 시험대상으로 명시된 펌프, 밸브, 방진기 등은 가동중시험 계획에 포함시켜야 한다.

MOA 1320 등급 분류

계통경계 내의 어느 기기를 설계시방서 등급보다 높은 등급으로 건조한 경우에도 계통의 전반적인 등급에 영향을 미치지 않아야 한다.

Figure 2: An example of a document illustrating the content structure and formatting used for guidelines and requirements in nuclear power plant operations. The content is categorized into different sections, including general information, application scope, and specific criteria.

Query		발전사업자의 책임에 대해 알려주세요
Length	chunk	00교정점검 및 시험계획의 수행에 사용하는 모든 계측기와 시험장비는 발전사업자의 행정절차서 또는발전사업자가 승인한 품질보증 계획에 따라 교정하고 관리하여야 한다.MOA 5000 (예 비) MOA 6000 (예 비) MOA 7000 (예 비) MOA 8000 (예 비) MOA 9000기록 및 보고MOA 9100적용범위기록보유 요건은 가동중시험 기술기준에서 요구하는 가동전,가동중 시험 및 점검에서 발생된 기록에 적용하여야 한다.MOA 9200요건MOA 9210발전사업자 책임(1)발전사업자는 가동중시험 기술기준의 요건을 만족하는 가동전,가동중 시험 및 점검 계획과일정을 작성하여야 한다.(2)발전사업자는 가동전,가동중 시험 및 점검 기록들을 작성,보관하여야 한다.MOA 9220작성(1)시험기록은 시험 및 점검 요건에 따라 작성하여야 한다.(2)계획은 표지에 다음을 명기하여야 한다.- 작성일자- 발전사업자 명칭 및 소재지- 발전소 명칭 및 소재지- 각 호기별 명칭- 상업운전 개시일자MOA 9230가동중 시험 및 점검 결과시험 및
	generation	발전사업자의 책임은 가동중시험 기술기준의 요건을 충족하는 가동전, 가동중 시험 및 점검 계획과 일정을 작성하고 실행하는 것입니다. 또한, 이러한 시험 및 점검 기록들을 작성하고 보관하는 책임이 있습니다. 이러한 기록들은 발전소의 안전성과 효율성을 보장하기 위해 중요한 역할을 합니다.
Semantic	chunk	RP-5271994년3판,1991년MOA 1400참조 기준 및 규격가동중시험 기술기준의 참조 기준과 규격의 개정일자 및 발행년도는 표MOA 1400과 같다.표MOA 1400참조 기준 및 규격MOA 1500발전사업자의 책임발전사업자는 다음 사항에 대하여 책임이 있다.(1)발전소 각 기기의 적절한 등급 결정(1)(2)등급별 시험 또는 점검 대상기기의 계통경계 선정(3)시험 및 점검 면제 대상기기 선정(4)시험 및 점검 수행에 적절한 접근로와 공간을 확보하도록 기기를 설계 및 배치부록B임의요건을 지침으로참조(5)시험 계획 및(6)시험 및 점검일정의 수립지침 및 절차서작성(7)발전사업자의 품질보증계획에 따라 점검 및 시험을 수행,평가하는 자의 자격인정(8)시험 및 점검 수행(9)평가 근거를 마련하고,향후 시험 또는 점검 결과와 비교가 가능하도록 결과를 기록(10)시험 및 점검 결과 평가(11)적절한 시험 및 점검 기록(시험 및 점검 결과 및 절차 기술) 유지(12)기기 또는 계통 수명기간 동안 시험 및 점검 결과 보존(13) KEPIC-QAP에 따른 품질보증계획 문서화MOA 1600접근성시험 및 점검 수행에 필요한 점검자 및 장비의 접근성이 확보되어야 한다.주(1)등급분류는 원자력안전위원회 고시 제2014-15(원자로시설의 안전등급과 등급별 규격에 관한 규정)에 따른다.MOA 2000용어정의가동전시험(PST) : 기기 설치 후부터 노심열에 의한 최초 전력생산 이전까지의 기간 또는 가동중 원전에서 기기 가동전 기간에 수행하는 시험가동전시험 기간 : 기기 설치
	generation	발전사업자는 다음 사항에 대해 책임이 있습니다: 1. 발전소 각 기기의 적절한 등급 결정 2. 등급별 시험 또는 점검 대상기기의 계통경계 선정 3. 시험 및 점검 면제 대상기기 선정 4. 시험 및 점검 수행에 적절한 접근로와 공간을 확보하도록 기기를 설계 및 배치 5. 시험 계획 및 시험 및 점검일정의 수립지침 및 절차서 작성 6. 발전사업자의 품질보증계획에 따라 점검 및 시험을 수행, 평가하는 자의 자격 인정 7. 시험 및 점검 수행 8. 평가 근거를 마련하고, 향후 시험 또는 점검 결과와 비교가 가능하도록 결과를 기록 9. 적절한 시험 및 점검 결과 평가 10. 기기 또는 계통 수명기간 동안 시험 및 점검 결과 보존 11. KEPIC-QAP에 따른 품질보증계획 문서화 이러한 책임은 발전소 기기의 안전성과 효율적인 운영을 보장하기 위해 필수적입니다.
StyleDFS	chunk	## MOA 1000 일반사항 ### MOA 1500 발전사업자의 책임 발전사업자는 다음 사항에 대하여 책임이 있다. (1) 발전소 각 기기의 적절한 등급 결정 (2) 등급별 시험 또는 점검 대상기기의 계통경계 선정 (3) 시험 및 점검 면제 대상기기 선정 (4) 시험 및 점검 수행에 적절한 접근로와 공간을 확보하도록 기기를 설계 및 배치 부록 B 임의요건을 지침으로 참조 (5) 시험 계획 및 (6) 시험 및 점검 일정의 수립 지침 및 절차서 작성 (7) 발전사업자의 품질보증계획에 따라 점검 및 시험을 수행 , 평가하는 자의 자격인정 (8) 시험 및 점검 수행 (9) 평가 근거를 마련하고 향후 시험 또는 점검 결과와 비교가 가능하도록 결과를 기록 (10) 시험 및 점검 결과 평가 (11) 적절한 시험 및 점검 기록(시험 및 점검 결과 및 절차 기술) 유지 (12) 기기 또는 계통 수명기간 동안 시험 및 점검 결과 보존 (13) KEPIC-QAP 에 따른 품질보증계획 문서화
	generation	발전사업자는 다음 사항에 대해 책임이 있습니다: 1. 발전소 각 기기의 적절한 등급 결정 2. 등급별 시험 또는 점검 대상기기의 계통경계 선정 3. 시험 및 점검 면제 대상기기 선정 4. 시험 및 점검 수행에 필요한 적절한 접근로와 공간을 확보하도록 기기를 설계 및 배치 5. 시험 계획 및 시험 및 점검 일정의 수립 지침 및 절차서 작성 6. 발전사업자의 품질보증계획에 따라 점검 및 시험을 수행, 평가하는 자의 자격 인정 7. 시험 및 점검 수행 8. 평가 근거를 마련하고 향후 시험 또는 점검 결과와 비교가 가능하도록 결과를 기록 9. 시험 및 점검 결과 평가 10. 적절한 시험 및 점검 기록(시험 및 점검 결과 및 절차 기술) 유지 11. 기기 또는 계통 수명기간 동안 시험 및 점검 결과 보존 12. KEPIC-QAP에 따른 품질보증계획 문서화

Table 9: Comparison of Generated Answers for Queries. “Chunk” indicates the source content, and “generation” denotes the generated results. Instruction tuned Llama-3-Open-Ko-8B model was used for generation.

Query		Please provide information on the responsibilities of the power plant operator
Length	chunk	All measuring instruments and testing equipment used for the calibration inspection and test plan must be calibrated and managed according to the administrative procedures of the power plant operator or the quality assurance plan approved by the power plant operator. MOA 5000 (reserve) MOA 6000 (reserve) MOA 7000 (reserve) MOA 8000 (reserve) MOA 9000 records and reports MOA 9100 scope The record retention requirements must apply to records generated from pre-operational and operational tests and inspections required by the in-service testing technical standards. MOA 9200 requirements MOA 9210 responsibilities of the power plant operator (1) The power plant operator must prepare a pre-operational and operational test and inspection plan and schedule that satisfies the requirements of the in-service testing technical standards. (2) The power plant operator must prepare and maintain records of pre-operational and operational tests and inspections. MOA 9220 preparation (1) Test records must be prepared according to the requirements for tests and inspections. (2) The plan must state the following on the cover - preparation date - name and location of the power plant operator - name and location of the power plant - name of each unit - commercial operation start date MOA 9230 pre-operational tests and inspection results of tests and
	generation	The responsibility of the power plant operator is to prepare and carry out a pre-operational and operational test and inspection plan and schedule that satisfies the requirements of the in-service testing technical standards. Additionally, it is responsible for preparing and maintaining records of these tests and inspections. These records play an important role in ensuring the safety and efficiency of the power plant.
Semantic	chunk	RP-527 1994 3rd edition, 1991 MOA 1400 reference standards and specifications The revision dates and publication years of the reference standards and specifications in the in-service testing technical standards are as follows Table MOA 1400. Table MOA 1400 reference standards and specifications MOA 1500 responsibilities of the power plant operator The power plant operator is responsible for the following: (1) determining the appropriate grade of each device in the power plant, (2) selecting the system boundaries of test or inspection target devices by grade, (3) selecting devices exempted from test or inspection, (4) designing and arranging devices to ensure appropriate access and space for carrying out tests and inspections, (5) preparation of test plans and procedures, (6) preparing test and inspection schedules and procedures, (7) recognizing the qualifications of those who perform tests and inspections according to the power plant operator's quality assurance plan, (8) conducting tests and inspections, (9) preparing evaluation grounds and recording results to compare with future tests or inspections, (10) evaluating test and inspection results (11) maintaining appropriate test and inspection records (describing test and inspection results and procedures), (12) preserving test and inspection results for the lifetime of the devices or systems, (13) documenting quality assurance plans according to KEPIC-QAP MOA 1600 accessibility Ensuring the accessibility of inspectors and equipment needed for conducting tests and inspections Note (1) The grade classification is according to Nuclear Safety and Security Commission Notice No. 2014-15 (Safety grades and specifications for reactor facilities). MOA 2000 definitions Pre-operational Test (PST): A test conducted during the period from the installation of a device until the first power production by core heat or during the pre-operational period in an operational nuclear power plant Pre-operational test period: Period from device installation
	generation	The responsibilities of the power plant operator are as follows: 1. Determining the appropriate grade of each device in the power plant, 2. Selecting the system boundaries of test or inspection target devices by grade, 3. Selecting devices exempted from tests or inspections, 4. Designing and arranging devices to ensure appropriate access and space for carrying out tests and inspections, 5. Preparation of test plans and procedures, 6. Preparing test and inspection schedules and procedures, 7. Recognizing the qualifications of those who perform tests and inspections according to the power plant operator's quality assurance plan, 8. Conducting tests and inspections, 9. Preparing evaluation grounds and recording results to compare with future tests or inspections, 10. Evaluating the test and inspection results 11. Maintaining appropriate test and inspection records (describing test and inspection results and procedures), 12. Preserving test and inspection results for the lifetime of the devices or systems, 13. Documenting quality assurance plans according to KEPIC-QAP These responsibilities are essential to ensure the safety and efficient operation of power plant devices.
StyleDFS	chunk	## MOA 1000 General Requirements ### MOA 1500 Responsibilities of the Power Plant Operator The power plant operator is responsible for the following: (1) determining the appropriate grade of each device in the power plant, (2) selecting the system boundaries of test or inspection target devices by grade, (3) selecting devices exempted from test or inspection, (4) designing and arranging devices to ensure appropriate access and space for carrying out tests and inspections, Appendix B reference of optional requirements (5) preparation of test plans and procedures, (6) preparing test and inspection schedules and procedures, (7) recognizing the qualifications of those who perform tests and inspections according to the power plant operator's quality assurance plan, (8) conducting tests and inspections (9) preparing evaluation grounds and recording results to compare with future tests or inspections (10) evaluating test and inspection results (11) maintaining appropriate test and inspection records (describing test and inspection results and procedures), (12) preserving test and inspection results for the lifetime of the devices or systems (13) documenting quality assurance plans according to KEPIC-QAP
	generation	The responsibilities of the power plant operator are as follows: 1. Determining the appropriate grade of each device in the power plant, 2. Selecting the system boundaries of test or inspection target devices by grade, 3. Selecting devices exempted from tests or inspections, 4. Designing and arranging devices to ensure appropriate access and space for carrying out tests and inspections, 5. Preparation of test plans and procedures, 6. Preparing test and inspection schedules and procedures, 7. Recognizing the qualifications of those who perform tests and inspections according to the power plant operator's quality assurance plan, 8. Conducting tests and inspections, 9. Preparing evaluation grounds and recording results to compare with future tests or inspections, 10. Evaluating the test and inspection results 11. Maintaining appropriate test and inspection records (describing test and inspection results and procedures), 12. Preserving test and inspection results for the lifetime of the devices or systems, 13. Documenting quality assurance plans according to KEPIC-QAP

Table 10: Comparison of Generated Responses for Queries. Translated in English using GPT-4 (OpenAI, 2024).