Ensuring Data Sovereignty in Knowledge Management: An Offline RAG Architecture for Critical Decision Support

Vitor Mathias da Costa, Gabriel de Sapienza Luna, Arthur Pinheiro de Araújo Costa, Allyson Andrade da Silva, José Victor de Pina Corriça, Leonardo Oldani Felix e Everton dos Santos do Nascimento

Centro de Instrução Almirante Alexandrino (CIAA), Marinha do Brasil, Rio de Janeiro/RJ – Brasil

***Abstract*— This paper presents a case study on "*RadRAG*," an offline Retrieval-Augmented Generation (RAG) architecture designed to ensure data sovereignty while enhancing Knowledge Management (KM) in critical defense environments. In the domain of radiological protection, professionals must navigate a complex and fragmented regulatory framework in which precision is essential. While Large Language Models (LLMs) offer transformative potential for knowledge retrieval, standard cloud-based implementations pose unacceptable risks to information security and data governance. To address this issue, we developed a fully local *RAG* system that operates on mid-range hardware, thereby democratizing access to advanced AI-driven decision support without compromising sensitive data. The system integrates a local Llama 3 model with a specialized vector retrieval pipeline grounded in the official normative corpus (CNEN standards), thereby mitigating hallucinations and ensuring source traceability. We validated this architecture through rigorous benchmarking of chunking strategies, semantic, hierarchical, and ensemble, against leading commercial models. The results demonstrate that the proposed local architecture, utilizing an ensemble retrieval strategy, achieved a competitive accuracy of 89.9%, confirming that locally hosted small LLMs can serve as effective, secure, and sovereign tools for critical organizational knowledge management.**

***Keywords*— Retrieval-Augmented Generation, Data Sovereignty, Radiological Protection**

I. INTRODUCTION

The field of radiological protection in the nuclear sector operates within a complex regulatory framework, characterized by a vast body of information distributed across numerous standards and guidelines, often written in highly technical language. In this rigorous operational environment, where precision is paramount, the manual consultation and interpretation of these documents can pose significant challenges, even for experts in the field. The search for specific information is not only time-consuming but also prone to errors, which may compromise regulatory compliance and, critically, nuclear and radiological safety for both personnel and the environment. Therefore, professionals must possess a strong command of this extensive regulatory corpus, necessitating effective tools that facilitate rapid and precise access to information, enabling navigation of the regulatory landscape with maximum efficiency and reliability.

V. M. Costa, vitor.mathias@marinha.mil.br; G. S. Luna, gabriel.luna@marinha.mil.br; A. P. Costa, arthur.costa@marinha.mil.br; A. A. Silva, allyson.andrade@marinha.mil.br; J. V. Corriça, corrica@marinha.mil.br; L. O. Felix, leonardo.felix@marinha.mil.br; E. S. Nascimento, santos.everton@marinha.mil.br. This work was carried out at CIAA.

In response to this challenge, this work presents the development of an advanced Retrieval-Augmented Generation (RAG) system designed to support information retrieval and decision-making in radiological protection within the Brazilian Navy (MB), an institution where precision, reliability, and data sovereignty are paramount. In sensitive environments, information flow cannot depend on external providers; therefore, a fully offline architecture has been adopted to minimize vulnerabilities and ensure source traceability.

The main objective of this research is to design and evaluate *RadRAG*, a local Python-based pipeline that integrates *ChromaDB* [1] for vector indexing and *llama3.1:8b* [2] as the generative model. Its effectiveness was assessed across various internal configurations, including variations in chunking strategies, hyperparameters, and reranker models, and benchmarked against leading large-scale commercial *LLM*s under a unified evaluation protocol. The central hypothesis is that a combination of effective document segmentation, hybrid retrieval, and cross-encoder-based reranking reduces inaccuracies and ensures that responses remain strictly aligned with the regulatory corpus.

The evaluation of the systems in this study follows the *LLM-as-a-judge* paradigm, which the scientific community is actively exploring. This method leverages high-capacity models, such as *ChatGPT*, to conduct automated evaluations [3]. Standardized guidelines have been established to ensure consistent judgments from the model, and this unified protocol has been applied to all *RadRAG* variants and external *LLMs*, ensuring the comparability of results.

To support this evaluation, a 30-question benchmark dataset was developed in collaboration with subject-matter experts (SMEs) in radiological protection within the Brazilian Navy. The dataset comprises questions of varying difficulty levels, specifically designed to challenge the system's retrieval capabilities, evidence aggregation, and source fidelity. The queries require, for example, the synthesis of information from multiple sources, the extraction of tabular data, and the integration of facts from different sections within the same standard, thereby increasing the complexity of retrieval and fidelity tasks.

The corpus includes standards from the Brazilian National Nuclear Energy Commission (CNEN), focusing on fundamental requirements and definitions regarding radiological protection. Specifically, the following standards were utilized: CNEN-NN-3.01, CNEN-NE-3.02, CNEN-NN-3.05, and CNEN-NN-7.01. [4].

In summary, this work presents a well-documented experimental implementation that aligns with the Brazilian Navy's (MB) requirements and introduces a quantitative and comparable performance evaluation metric.

II. RELATED WORK

The chunk construction stage in RAG systems has been receiving special attention from numerous researchers, as it is an indispensable pillar of the architecture. It serves the primary function of decomposing unstructured documents into smaller, cohesive, and semantically rich units of information.

According to Lu et al. [5], LLM-guided hierarchical approaches, such as HiChunk, represent an advanced method that interprets and structures document content into sections and subsections, organizing chunks at multiple levels while preserving cohesion. Concurrently, Huang et al. [6] discuss semantic similarity-based splitting as an excellent alternative to fixed-size chunking. This approach is also utilized by Zhong et al. [7] in the construction of SemRAG, which employs a semantic chunking algorithm to group sentences based on embedding similarity.

In this context, our work draws inspiration from the cited approaches while introducing key distinctions. Specifically, we do not employ an LLM to determine chunk boundaries in hierarchical modeling; instead, we evaluate an ensemble-based technique. Furthermore, for the recall component, this study is inspired by the hybrid approach proposed in Blended RAG [8], which performs a combined semantic and lexical search and retrieves the most relevant chunks through weighted fusion.

A. Advantages of RAG over pure LLMs

LLMs, while highly powerful, exhibit significant limitations in technical domains [9]. These limitations arise because their knowledge is inherently static and because they are susceptible to hallucinations, the generation of factually incorrect yet plausible information. This propensity for error is further exacerbated by an intrinsic lack of traceability, which makes it difficult to verify sources and validate generated outputs [6].

To mitigate these issues, the *Retrieval-Augmented Generation* (RAG) architecture was developed, in which the system operates in two stages. First, a retrieval component searches an external database for the information snippets that are most relevant to the user's query. Next, in the augmented generation stage, the system combines this retrieved context with the original prompt, instructing the *LLM* to formulate a response grounded in the evidence found [10].

This approach offers crucial advantages over traditional *LLM*s, as it drastically reduces hallucinations by anchoring the model to a factual, domain-specific body of knowledge [6]. In addition, *RAG* overcomes the static-knowledge barrier and introduces transparency by allowing citation of the sources used, enabling the user to verify the information [10]. Consequently, *RAG* emerges as a more efficient and promising solution for building Artificial Intelligence (AI) assistants specialized in domain-specific tasks [11].

III. METHODS AND HARDWARE EMPLOYED

This section describes the theoretical framework underpinning the work, the configurations adopted for *RadRAG*, the computational environment utilized, and the costs associated with the experiment.

A. Hardware Employed

The experiment was conducted on a commercial laptop equipped with an Intel Core i5-12500H processor, 16 GB of RAM, and an NVIDIA GeForce RTX 3050 Ti laptop GPU. Essential resources consisted of open-source software libraries and pre-trained models, all of which are publicly available. It must be emphasized that the *RAG* program was developed to operate entirely in local mode. This constitutes its primary advantage, as it requires no internet access for its functionality.

B. System pipeline

The first fundamental stage of a *RAG* system's architecture is the indexing and storage phase [12], depicted as stage 2.0 in Figure 1, in which a dataset is converted into an optimized, semantically searchable corpus.

In the *RadRAG* pipeline, the process begins with the ingestion of Portable Document Format (PDF) documents via *PyMuPDF*. This library is responsible for parsing the document and segmenting it into sections and subsections while preserving the material's logical structure, which corresponds to the *Source* phase in stage 2.0 of Figure 1. Useful metadata, such as the standard's name and title, are added to each section and later inherited by the chunks. This metadata is of utmost importance, as it is crucial for citation generation and contextual retrieval. Next, the documents are processed and transformed into chunks during the *Transform* step, then converted into embeddings by the `BAAI/bge-m3` model and indexed in the *ChromaDB* vector database.

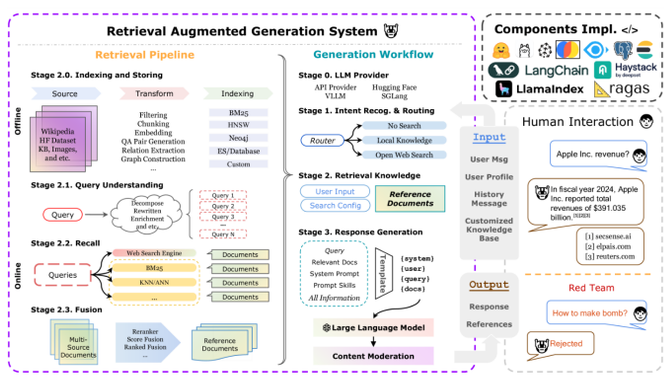

Fig. 1

OVERVIEW OF A RETRIEVAL-AUGMENTED GENERATION (RAG) SYSTEM, INCLUDING INDEXING, RETRIEVAL, GENERATION, AND HUMAN-IN-THE-LOOP COMPONENTS [12].

After indexing, the system is activated by a user prompt (stage 1), followed by query understanding (stage 2.1) and recall (stage 2.2), which retrieves candidate documents most likely to contain the answer [12].

RadRAG adopts a hybrid recall strategy inspired by Blended RAG [8]: semantic similarity search (HNSW) is executed in parallel with lexical matching (BM25). Candidates from both branches are pooled and deduplicated; their scores are normalized and combined through weighted fusion to produce

a final ranking score (Equation (1)). The candidates are then pre-ranked based on this score and forwarded to the reranker.

$$\text{score}_{\text{final}} = \alpha\, \text{score}_{\text{sem}} + (1-\alpha)\, \text{score}_{\text{lex}}, \tag{1}$$

Subsequently, to refine this selection, a re-ranking component comes into play to reorder the candidates based on a more fine-grained relevance analysis (the *reranker*, referenced at stage 2.3 in Figure 1)—in *RadRAG* we used a *cross-encoder* tool. This process ensures that only a more relevant subset is forwarded to the *LLM* (reference documents at stage 2.3 in Figure 1). This step is essential to mitigate the risk of *LLM* hallucinations and to optimize the response context [12].

Finally, the flow proceeds to answer generation (denoted as stage 3 in Figure 1), where the *LLM* receives an augmented prompt composed of several inputs, such as the user's original query, the selected reference documents, and the system prompt that is preconfigured by the developer. Based on this set of information, the model synthesizes and produces a final response (output) to be presented to the user [12].

C. Types of chunking used

The chunk construction step is an essential pillar in the architecture of a *RAG* system, since improper segmentation can lead to loss of context or the insertion of noise into the retrieval process, [13] [6]. Given its criticality, this research evaluated *RadRAG*'s performance across multiple strategies to determine the most effective one.

An advanced approach is hierarchical chunking, which can leverage an *LLM* to interpret and structure content at multiple levels of granularity, as in the *HiChunk* method [5]. However, due to its high computational cost, *RadRAG* implemented a simplified adaptation of this method.

The decision not to employ an *LLM* for the dynamic determination of breakpoints, unlike approaches such as *HiChunk*, is strictly grounded in operational efficiency and hardware resource constraints. The use of language models for segmentation imposes significant computational overhead and increases data ingestion latency. Given that *RadRAG* was designed to operate on mid-range commercial hardware (specifically an *NVIDIA GeForce RTX 3050 Ti GPU*) and in a fully offline environment, a structural approach based on table of contents and formatting analysis via *PyMuPDF* was chosen. Despite this adaptation, the core idea from reference works was incorporated: the system performs vector search by identifying smaller, semantically dense chunks (children), using them as anchors to retrieve the larger chunks (parents) to which they belong. This strategy ensures precision in similarity retrieval while preserving the extended context necessary for response generation, mitigating information loss without incurring the computational cost of AI-based segmentation.

Another fundamental technique is semantic chunking, which groups sentences based on the similarity of their embeddings, preserving thematic coherence. This approach is viewed as more advantageous than fixed-size chunking, which often fails to capture the text's logical structure [6]. Grouping is performed using the cosine distance metric, according to the grouping rule defined by function g in Equation (2) [7], to identify clusters of semantically close sentences [7].

$$g = \left\{ c_i \;\middle|\; \delta\big(d(c_i), d(c_{i+k})\big) = 1 - \frac{d(c_i)\cdot d(c_{i+k})}{\|d(c_i)\|_2 \|d(c_{i+k})\|_2} < \tau,\ \forall k \in [0, n] \right\}. \tag{2}$$

The grouping function g, formalized in Equation (2), defines the logical rule for consolidating semantic blocks. In this expression, c_i represents the pivot sentence under analysis, while c_{i+k} denotes the adjacent sentences within a context window. The function d denotes the mapping of the text to its respective embedding vector. The merging criterion is based on the cosine distance, calculated by subtracting the cosine similarity (the fractional term in the equation) from 1. Thus, the algorithm determines that neighboring sentences are aggregated into the same chunk only if the vector distance between them is below the established similarity threshold, ensuring high thematic cohesion in the resulting segment.

We also evaluated a hybrid strategy that combines hierarchical boundaries with semantic micro-segmentation. Parent chunks defined by PyMuPDF's section structure are further partitioned into thematically coherent child chunks, which are persisted for the recall stage.

The best-performing configuration, and the one adopted in *RadRAG*'s final version, is an *ensemble* of hierarchical and semantic chunking. Both strategies execute independently over the same source, and all resulting chunks are indexed. While this increases redundancy, it materially improves the likelihood of retrieving evidence well aligned with the user's query, a justified trade-off in accuracy-critical settings.

D. Hyperparameters used in the system

The architecture and behavior of *RadRAG* are defined by a set of core hyperparameters, summarized in Table (1). In the advanced-chunking module, we expose size and overlap settings for parent chunks (parent-chunk-size, parent-overlap) and child chunks (child-chunk-size, child-overlap), which determine the granularity of the hierarchical strategy. Additionally, similarity-threshold calibrates grouping in semantic segmentation, while min-chunk-size sets a lower bound for generated chunk length.

For the *ChromaDB* vector database, the system employs a hybrid search with a weight of 0.7 for the semantic component (the value α in Equation 1) and 0.3 for the lexical component, using cosine distance as the similarity metric.

In the answer-generation stage, *RadRAG* uses the *llama3.1:8b* model, selected for enabling fully offline operation and its low computational cost. The generation temperature is set to 0.1 to reduce randomness and the propensity for hallucination, while max_tokens limits the output length to 1000 tokens. In addition, text vectorization is performed by the *BAAI/bge-m3* embedding model.

IV. Analysis of Results

This section describes the evaluation technique used to validate the study and presents the benchmark comparing different *RadRAG* versions, followed by a comparison between the best-performing version and commercial *LLMs*.

A. LLM-as-judge

Validating the accuracy of complex systems such as *RAG* is a methodological challenge and has traditionally relied on

human evaluation. In this context, the research community has actively explored the approach known as LLM-as-judge, in which high-capacity models (e.g., *ChatGPT*) are employed to conduct automatic evaluations [3]. In this study, we adopted this technique and used *NotebookLM* as the judge, due to its strong performance in locating answers within documents.

TABELA I

RadRAG HYPERPARAMETERS.

Component	Parameter	Value
advanced_chunking	parent_chunk_size	800
	parent_overlap	200
	child_chunk_size	200
	child_overlap	50
	min_chunk_size	50
chromadb	distance_function	cosine
	search_type	hybrid
	search_semantic_weight (α)	0.7
	search_keyword_weight ($1-\alpha$)	0.3
llm	model	llama3.1:8b
	temperature	0.1
	max_tokens	1000
embeddings	model	BAAI/bge-m3
reranker	model	BAAI/bge-reranker-v2-m3
	top_k_rerank	20
	top_k_final	4

Note: α is the semantic weight from Eq. 1.

Additionally, to examine potential bias associated with a single evaluator, the responses generated by the best-performing *RadRAG* configuration were also evaluated by *GPT, Gemini, and DeepSeek*. All judges received the same instruction prompt, grading criteria, and experts' reference answers.

To structure the LLM-based evaluation, we defined a standardized experimental protocol. First, a questionnaire was developed with the assistance of radiation-protection specialists, thereby creating an answer key to serve as ground truth for the judge. Next, we defined a detailed set of grading criteria that *NotebookLM* was instructed to follow. Under these criteria, each question was assigned a maximum score of 5.0 points, split between technical content (4.7 points) and the accuracy of the bibliographic reference (up to 0.3 points), and a penalty of 0.5 points was applied to any response that exhibited hallucination.

The protocol was orchestrated by a detailed instruction prompt that defines *NotebookLM's* role as an objective academic evaluator and requires strict adherence to the grading criteria, emphasizing the principle of semantic equivalence—that is, assessing the meaning of the answer rather than only literal matching. Additionally, the prompt establishes a two-phase workflow: the first phase compares the grading criteria with the candidate answers; the second verifies any incorrect information (hallucinations) and assigns a final score to each question.

In this way, each model's final accuracy was computed analogously to a written exam: the total score obtained by the system across the 30 questions was divided by the maximum possible score (150 points). The result of this division, expressed as a percentage, represents the model's accuracy and provides a quantitative, comparable performance metric. Each *RadRAG* configuration reported in Table II was evaluated in a single run over the 30-question benchmark. Because both answer generation and LLM-based judging are stochastic, the reported results should be interpreted as point estimates.

B. Comparison among RadRAG versions

To determine the most effective architecture for *RadRAG*, we conducted a systematic comparative analysis across the different system versions, detailed in Table 2. This analysis varied three main independent variables: (a) chunking strategy semantic, hybrid, or ensemble (semantic + hierarchical); (b) reranker model BAAI/bge-reranker-v2-m3 or cross-encoder/ms-marco-MiniLM-L-6-v2; and (c) the top_k_rerank hyperparameter, using values 20 and 50.

TABELA II

COMPARISON OF *RadRAG* VERSIONS.

Setting	Values (v1 $\rightarrow$ v5)
Advanced chunking strategy	v1: semantic + hierarchical v2: semantic + hierarchical v3: semantic v4: hybrid v5: semantic + hierarchical
top_k_rerank	v1: 20; v2: 50; v3: 20; v4: 20; v5: 20
top_k_final	v1: 4; v2: 4; v3: 4; v4: 4; v5: 4
Reranker model	v1–v4: BAAI/bge-reranker-v2-m3 v5: cross-encoder/ms-marco-MiniLM-L-6-v2
Accuracy	v1: 89.9%; v2: 89.2%; v3: 70.8%; v4: 63.8%; v5: 63.6%
Processing time	v1: 36 min; v2: 108 min; v3: 41 min; v4: 15 min; v5: 15 min

The ensemble chunking approach, which combines hierarchical and semantic methods independently, was the cornerstone of *RadRAG*'s final version (*RadRAG* v1). Moreover, this strategy produced a diverse repository of 884 chunks (572 hierarchical and 312 semantic) and was validated as the most robust. Thus, the increased storage consumption and chunk redundancy are justified by the performance gains delivered by this approach.

The superiority of this strategy was demonstrated in direct comparisons: *RadRAG* achieved performance that was 19.1 percentage points higher than the purely semantic approach (*RadRAG* v3). The difference was even more pronounced relative to the hybrid strategy (*RadRAG* v4), with a margin of 26.1 percentage points. In addition, that strategy—besides being less effective—exhibited critical failures such as hallucinations and formatting errors, and was therefore deemed ineffective in this study.

The choice of re-ranking components also proved decisive: selecting a cross-encoder model was crucial to the success of this research and yielded a 26.3 percentage-point improvement compared with *RadRAG* v5.

Additionally, a sensitivity analysis was performed on the top-k-rerank hyperparameter (*RadRAG* v2), showing that increasing the number of re-ranked chunks from 20 to 50 led to roughly three times longer processing time without a meaningful accuracy gain. Hence, the results were considered inconclusive—this small difference may be attributable to variability in the LLM-as-judge evaluation. Therefore, the value 20 was kept for the remaining comparisons as the best trade-off between computational cost and effectiveness.

The systematic evaluation of *RadRAG* variants validated the highest-performing architecture, revealing that the ensemble chunking strategy and a high-performance reranker model contributed most to the system's accuracy and robustness. Consequently, the final version of *RadRAG* produced high-quality answers that systematically included bibliographic references.

For the best-performing *RadRAG* configuration, the cross-evaluation yielded aggregate scores of 89.9%, 91.9%, 91.6%, and 90.6% for *NotebookLM, GPT, Gemini, and DeepSeek*, respectively. Detailed question-by-question results are available in the online supplementary table: `https://docs.google.com/spreadsheets/d/1vE_nney8gkbQenmy-SL7MJZozzoItJxHo2KEh_1b0JE/edit?usp=sharing`.

Based on these data, it can be concluded that, although the *LLM-as-a-judge* evaluation approach is susceptible to the inherent biases of each model, the observed agreement among evaluators provides evidence of evaluation consistency for the best-performing *RadRAG* configuration. Because the cross-evaluation was not applied to all five configurations, it does not constitute a complete validation of the robustness of LLM-based judging across the experiment, despite the observed consistency. Accordingly, *NotebookLM* was retained as the standard evaluator for the responses generated by the remaining system versions and the commercial models compared in this study.

C. Comparison of RadRAG versions with commercial LLMs

To contextualize *RadRAG*'s performance, we benchmarked the system against commercially available (and free) *LLMs*. The evaluation methodology consisted of starting a new conversation session for each *LLM*, providing this work's reference publications as the initial context, and then submitting the set of 30 questions with the instruction to generate answers including proper bibliographic references.

The *DeepSeek-V3* model posed a particularity during the study: its context limitation prevented loading all four publications simultaneously. The evaluation was therefore split into two separate conversations—one containing the CNEN-NN-3.01 standard and another with the remaining publications. It is worth noting that this segmentation reduced the complexity and amount of information in each query's context, which may have facilitated answer retrieval and contributed to its superior performance.

In the overall ranking, *RadRAG* placed second, surpassed only by *DeepSeek-V3*. The system outperformed large, high-development-cost models such as *Gemini 2.5 Flash, Grok 4 Fast, GPT-5*, and a Python-based *RAG* system using an *OpenAI API* key with the *gpt-4o-mini* model.

As illustrated in Figure 2, *RadRAG*'s performance was highly competitive, positioned just 3.7 percentage points below the top-performing model. This result validates *RadRAG* as a high-accuracy alternative when compared with leading market models. Its chief advantage, however, lies in the ability to operate fully offline, ensuring data sovereignty and security.

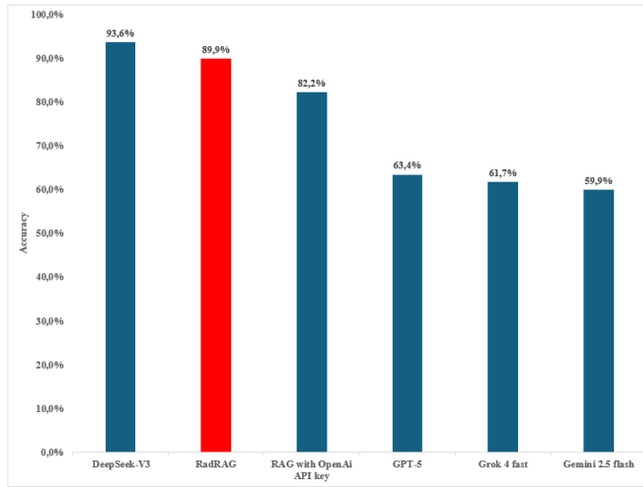

Fig. 2
MODEL ACCURACY COMPARISON.

V. CONCLUSION

This work was motivated by the need for a specialist virtual assistant in radioprotection that is simultaneously accurate, reliable, and secure for the Brazilian Navy's work environment, culminating in the development of *RadRAG*. The proposed objectives were met, resulting in a locally operated tool, implemented in Python, that uses a *RAG* architecture with a ChromaDB vector store and the *llama3.1:8b LLM*. The system proved capable of generating answers grounded exclusively in the provided documentary sources, fulfilling the fundamental requirement of mitigating hallucinations and ensuring data privacy.

The experimental validation revealed significant technical contributions: the ensemble chunking architecture (semantic + hierarchical) proved substantially superior to conventional approaches, surpassing the purely semantic strategy by 19.1 percentage points and the hybrid strategy by 26.1 percentage points. Moreover, the study highlighted the criticality of the reranker choice: selecting a high-performance cross-encoder (BAAI/bge-reranker-v2-m3) yielded a 26.3 percentage-point accuracy gain. These results show that combining diversified segmentation with precise re-ranking is essential to the system's success.

In a competitive landscape, *RadRAG* exhibited notable performance, outperforming commercial *LLMs* such as Gemini, GPT-5, and Grok, and standing only 3.7 percentage points below the best model evaluated, which benefited from a methodological advantage. Crucially, *RadRAG* achieves this level of performance while operating fully offline. For the Brazilian Navy—where information security and confidentiality are essential—this capability is strategically invaluable, eliminating risks associated with exposing data to third-party APIs.

In sum, this research not only delivered a functional AI system but also validated an advanced, secure, and high-performance *RAG* architecture.

A. Future work

Building directly on our findings, we outline the following directions to strengthen *RadRAG* while preserving its offline, security-oriented deployment:

(i) Domain adaptation of the LLM. Pursue lightweight fine-tuning (e.g., instruction/SFT or adapter-based LoRA) on radioprotection corpora to specialize the local model. Compare the current *llama3.1:8b* with larger/local variants (e.g., llama3.1:405b) under the same offline constraints, reporting accuracy–latency–memory trade-offs.

(ii) Retrieval and chunking. Evolve from a fixed fusion weight to a *query-adaptive* weighting (α) and adaptive candidate counts (dynamic *top_k_rerank*) based on query difficulty. Extend the ensemble strategy with learned boundaries: employ an agent to propose segmentation points with hierarchical awareness in the spirit of HiChunk [5], and study incremental indexing, duplicate control, and vector compression to offset the storage overhead observed with the ensemble.

(iii) Re-ranking. Evaluate next-generation cross-encoders (*e.g., BAAI/bge-reranker-v2.5-gemma2-lightweight*) as drop-in replacements, and investigate learning-to-rank with expert annotations. Use the current judge outputs as weak labels to warm-start re-ranking, then refine with curated pairs from domain specialists.

(iv) Grounded generation and citation control. Tighten grounding by enforcing passage-level attribution (page/section) and adding a post-generation verifier that rejects unsupported claims, flags formatting issues, and re-queries when evidence is insufficient. Couple this with stricter citation templates to reduce the residual hallucination/formatting errors identified in the baseline comparisons.

(v) Evaluation protocol.

Conduct repeated runs, particularly for the best-performing configuration, to quantify run-to-run variability. Complement the LLM-as-a-judge setup with a *multi-judge* scheme and targeted human adjudication on disagreements; report variability via bootstrap confidence intervals. Include ablations of α, chunking variants, and re-ranking windows, as well as stress tests with context truncation (mirroring the DeepSeek-V3 condition) to measure robustness. The present evaluation was limited to the hardware configuration described in this study and to a corpus composed of four CNEN standards. Future studies may also explore hardware with different computational capacities and broader, more heterogeneous normative corpora to further assess the generalizability of the observed results.

(vi) Deployment and operations. Introduce response caching and index refresh policies for normative updates, plus telemetry (offline) for latency, recall@k, and citation validity. These additions can improve responsiveness and maintainability without relaxing the offline and data-sovereignty requirements.

Overall, the most promising path is to pair the current ensemble chunking with query-adaptive retrieval and a stronger cross-encoder, while upgrading the evaluation pipeline to provide tighter uncertainty estimates and actionable diagnostics for future iterations.

REFERENCES

[1] ChromaDB, "Chroma: The ai-native open-source vector database," 2024, acessado em: 2024-03-30. [Online]. Available: https://www.trychroma.com

[2] M. AI, "Llama 3.1: Our most capable models to date," 2024, acessado em: 2024-07-23. [Online]. Available: https://ai.meta.com

[3] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, no. 1, 2023.

[4] Brasil, "Grupo 3 - proteção radiológica," https://www.gov.br/cnen/pt-br/acesso-rapido/normas/grupo-3, Comissão Nacional de Energia Nuclear (CNEN), n.d., [Online].

[5] W. Lu, K. Chen, R. Qiao, X. Sun, "Hichunk: Evaluating and enhancing retrieval-augmented generation with hierarchical chunking," *arXiv preprint arXiv:2509.11552*, 2025.

[6] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, 2025.

[7] K. Zhong, B. Suleiman, A. Erradi, S. Chen, "Semrag: Semantic knowledge-augmented rag for improved question-answering," *arXiv preprint arXiv:2507.21110*, 2025.

[8] K. Sawarkar, A. Mangal, S. R. Solanki, "Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers," in *2024 IEEE 7th international conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2024, pp. 155–161.

[9] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, Q. Li, "A survey on rag meeting llms: Towards retrieval-augmented large language models," in *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 6491–6501.

[10] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[11] M. Arslan, H. Ghanem, S. Munawar, C. Cruz, "A survey on rag with llms," *Procedia computer science*, vol. 246, pp. 3781–3790, 2024.

[12] A. Gan, H. Yu, K. Zhang, Q. Liu, W. Yan, Z. Huang, S. Tong, G. Hu, "Retrieval augmented generation evaluation in the era of large language models: A comprehensive survey," *arXiv preprint arXiv:2504.14891*, 2025.

[13] B. Smith A. Troynikov, "Evaluating chunking strategies for retrieval," Technical Report. Available online: https://research. trychroma. com . . . , Tech. Rep., 2024.