76th International Astronautical Congress, Sidney, Australia, 29 September - 3 October 2025.

IAC-25-D5-2-8-x100586

Language Modeling and Retrieval-Augmented Generation Integration in Space Operation Decision Support Tools

Ruben Belo[a]*, **Cláudia Soares**[a], **Marta Guimarães**[b]

[a] *NOVA LINCS, Computer Science Department, NOVA School of Science and Technology, Universidade NOVA de Lisboa* E-mail: rc.belo@campus.fct.unl.pt, claudia.soares@fct.unl.pt

[a] *Neuraspace, Portugal,* E-mail: marta.guimaraes@neuraspace.com

* Corresponding author

Abstract

The rapid expansion of the space industry has led to an increasing volume of technical documents, operational guidelines, and scientific publications. As space activities become more complex, effective decision-making and operations management require efficient processing of vast and ever-changing information sources to support new developments in the sector. Space operators must navigate regulations, guidelines, safety protocols, and mission-critical data, which makes knowledge retrieval a significant challenge. The ability to extract, synthesise, and communicate relevant information quickly is essential for ensuring safety, optimizing operations, and addressing emerging challenges in space traffic management and debris mitigation. To address these challenges, we propose a conversational agent powered by a Retrieval Augmented Generation (RAG) pipeline, leveraging Large Language Models (LLMs) and information retrieval techniques. Our system processes extensive documentation, retrieving and generating contextually relevant responses based on official reports, scientific literature, operational directives, and guidelines to provide actionable insights for space operations. Our methodology involves evaluating various retrieval strategies and embedding techniques to improve information accuracy. Furthermore, we explore different RAG configurations to optimize response generation, ensuring precise and reliable communication. To mitigate the risk of generating misinformation and harmful content, we integrate state-of-the-art techniques for conditioning LLM outputs, reinforcing safety and ethical considerations in our conversational agent's responses. This research integrates state-of-the-art AI technologies with domain-specific space knowledge to contribute to the development of intelligent systems for space operations. Our approach aims to improve decision-making and space traffic management, and address the growing challenges posed by space debris. Ultimately, this research supports the advancement of safer and more sustainable space activities and applications, aligning with global efforts to ensure long-term space exploration.

1. Introduction

As space missions increase in both complexity and frequency, the volume of documentation that engineers and operators must process has grown significantly. This includes engineering reports, operational guidelines, regulatory protocols, and scientific publications. Today's mission teams are required to make rapid, high-stakes decisions based on highly specialized and constantly ever-evolving information. However, much of this workflow still relies on manual analysis, legacy systems, and human recall, which makes the process inefficient, hard to scale, and costly [1]. These challenges render the task time-consuming, error-prone, and mentally demanding for mission personnel [1, 2].

An example of such documentation is the Concurrent Design Facility (CDF) report [3], typically 200–300 pages long, covering a broad spectrum of mission-related content. These reports frequently address topics such as mission design, system integration, payload components, ground infrastructure, operational planning, and risk evaluation [3]. Another use case was identified at the German Space Operations Center (GSOC) [2] of the German Aerospace Center (DLR), where operators must interpret large volumes of technical data and respond rapidly to unexpected events. Integrating LLM-based tools into this context offers a way to ease that burden, providing fast, context-aware responses that can assist engineers during both training and live missions [2].

These examples highlight a growing demand for intelli-

gent systems that can effectively navigate domain-specific documentation and provide accurate, context-aware insights in real time. As space operations become increasingly data-driven, the ability to quickly retrieve and synthesize relevant information is no longer optional, it is essential for mission success, operational safety, and long-term sustainability [3, 2, 1].

LLMs, particularly when combined with retrieval techniques through the RAG methodology, offer a promising solution to this challenge [2].

2. Related Work

The use of LLMs in space systems has gained attention as agencies seek to manage growing volumes of documentation. Early efforts adapted general-purpose models like BERT [4] to domain-specific contexts. To address limitations on specialized corpora, Berquand et al. [5] proposed SpaceTransformers, three models further pre-trained on curated space datasets.

Garcia-Silva et al. [3] introduced SpaceQA, the first open-domain QA system for space mission design, combining dense retrieval with neural reading. While effective, it highlighted the need for domain-specific fine-tuning. Another line of research explored question answering over structured knowledge bases, where natural language queries are mapped into structured database programs [6].

A multimodal approach for space robotics fine-tuned a vision-language model on Martian terrain, showing the potential of foundation models for scalable and autonomous operations [7]. More recently, LLMs have been evaluated in mission operations: at GSOC for information retrieval and decision support under time pressure [2], and as expert agents for mission management with capabilities like data analysis, summarization, and planning, designed for offline deployment to ensure confidentiality [1]. Some of this work [1, 2] also highlights the need to integrate external knowledge into LLMs. Retrieval Augmented Generation (RAG) has become a widely adopted approach [8] to improve factual accuracy and domain coverage.

2.1 Our Work

In this study, we present a evaluation of key components in RAG pipelines, applied to space-specific text. We evaluate both the retrieved context as well as quantify how retrieval quality influences the quality of said answers.

Our main contributions are:

1. Construction of a synthetic dataset with 7,036 question across five tasks: direct and abstract Q&A, fact-checking, summarization, and analysis (Sec. 4.1).
2. Examination of LLM preferences for passages obtained directly from retrieval and after reranking, to assess the quality of the retriever and the necessity of reranking (Sec. 5).
3. End-to-end evaluation of the RAG pipeline on the five tasks, to analyze how retrieval quality affects downstream generation and the model's robustness to irrelevant in-domain and out-of-domain passages within the context (Sec. 6).

Finally, it is important to note that much of the data used in mission operations is highly confidential and cannot be processed through cloud-based services [2]. To address this constraint, our work focuses on open-source models and frameworks that can be deployed entirely within closed, secure environments, aligning with the strict privacy and operational control both on the ground and in space.

3. Retrieval Augmented Generation: Enhancing LLMs with External Knowledge

With the widespread adoption of LLMs [9, 10, 11], these models have demonstrated exceptional capabilities across various linguistic tasks. However, they also encounter significant challenges in domain-specific or knowledge-intensive contexts [12]. Common issues include the generation of incorrect information, commonly called hallucinations [13]. To address these limitations, RAG integrates external knowledge as context to improve the performance of LLMs. The basic version of a RAG system consists of two tightly coupled components (see Fig.1): 1. A **Retriever** (Fig.1), which takes the user's query, performs vectorization, and searches for semantically relevant document chunks (passages) from the indexed vector store. This stage may also involve additional post retrieval steps such as reranking and metadata filtering to refine the selection of passages. 2. A **Generator** (Fig. 1), typically a LLM, which receives both the query and the retrieved passages as input. By incorporating these passages into its prompt, the generator produces a more accurate, contextually grounded, and informed response.

The key difference between generating an answer with and without RAG lies in the added context, which enables the LLM to incorporate external, up-to-date information into its response, improving the accuracy and credibility of the generated response [8].

3.1 The Retriever

The retriever serves as a bridge between the LLM and external knowledge sources, extracting relevant informa-

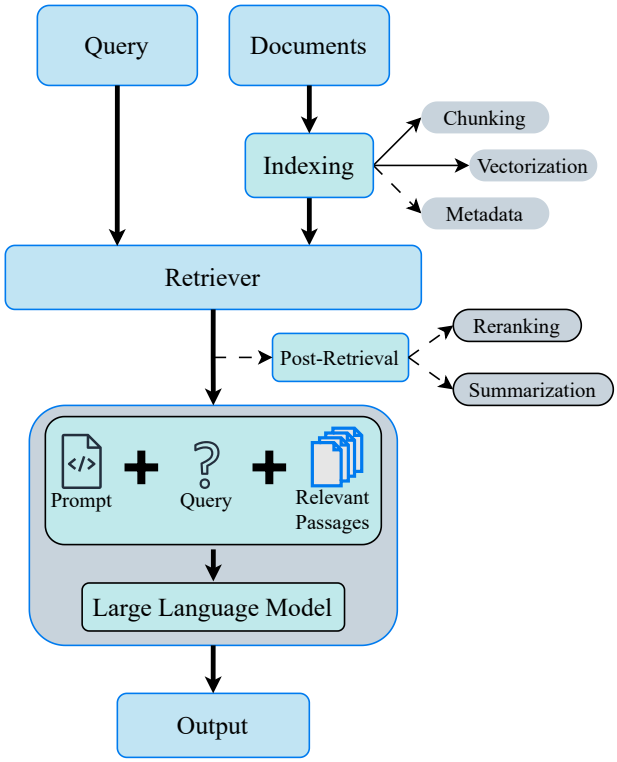

Fig. 1: RAG system architecture. Solid lines represent core components including query processing, document chunking, vectorization, retrieval, and language model integration. Dashed lines indicate optional components such as metadata filtering, reranking, and summarization steps.

tion to support generation. Retrieval quality relies heavily on semantic similarity, typically measured via cosine similarity between query and document embeddings [8]. These embeddings can be either sparse vectors (e.g., TF-IDF, BM25 [14]) or dense vectors from Deep Learning (DL) models. Hybrid retrieval, which combines both, often improves performance by leveraging complementary strengths [15].

Dense models can struggle in specialized domains like healthcare, law, finance, or space debris mitigation, where domain-specific jargon diverges from pretraining data [15]. To address this, approaches like hybrid search and learnable sparse models (e.g., SPLADE [16], BGE-M3 [17]) are emerging as effective alternatives.

3.2 The Generation

The generator, typically a LLM, produces the final response using both the input query and the retrieved context. Its effectiveness depends not only on the model architecture but also on its ability to integrate external information into coherent and grounded outputs. Fine-tuning can improve domain alignment and factual accuracy [18], while prompt-based methods such as few-shot learning and prompt engineering offer flexible adaptation to new tasks without retraining [8, 19]. However, both approaches are limited by context length and task complexity.

4. Data

A central objective of this work is to systematically evaluate the components of a RAG pipeline within the context of space debris mitigation. To support this evaluation, we required a dataset of domain-specific question. Since constructing such a dataset entirely by hand would be prohibitively resource-intensive, we adopted a hybrid strategy that combines expert-curated samples with synthetic data generation techniques.

We collected 39 documents from diverse sources, including official ESA reports and peer-reviewed conference papers on space debris mitigation. All documents are available in our GitHub repository[1]. These documents exhibit significant variation in format: some use one column, others two; they include diverse structural elements, headers, and numerous images. Such heterogeneity required advanced parsing tools capable of handling diverse element types and multiple PDF formats, including scanned documents. While these tools perform remarkably well on complex structures such as tables and equations, they often fail to consistently extract clean, unformatted text, resulting in incoherent sections and misplaced paragraphs.

Since our documents are text-based PDF files, and to address the previous errors, we used `semchunk` [20] to segment the documents into coherent passages, each with a maximum length of 2,000 tokens. While `semchunk` does not offer the full range of functionalities provided by more complex parsers, it focuses solely on textual content and segments the documents based on semantic coherence, resulting in well-structured passages.

Next, we need to generate questions for each chunk using synthetic data generation. We describe this process and the techniques used below.

[1] `https://github.com/rcBelo/Space-LM`

Table 1: Relative frequency (%) of relevance scores for Retriever and Reranker outputs using 2,000-token setting. Best values per Top-K (across all models) are in **bold**, second-best are underlined.

Top-K	Method	BM25				Qwen2-1.5B			
		0 ↓	1 ↓	2 ↑	3 ↑	0 ↓	1 ↓	2 ↑	3 ↑
Top-3	Retriever	2.12	3.44	**54.78**	39.66	1.73	2.44	53.35	42.47
	Reranked	1.29	1.78	52.18	44.76	**1.26**	**1.37**	52.41	**44.95**
Top-5	Retriever	2.98	4.82	**59.96**	32.24	2.49	3.38	59.16	34.97
	Reranked	2.13	2.76	58.55	36.56	**2.07**	**2.41**	58.62	**36.90**
Top-7	Retriever	3.78	5.82	**62.52**	27.88	3.15	3.91	62.17	30.77
	Reranked	2.65	3.70	62.04	31.61	**2.62**	**3.21**	62.14	**32.03**
Top-10	Retriever	4.80	7.05	64.38	23.76	3.85	4.74	64.72	26.68
	Reranked	3.51	4.80	64.87	26.82	**3.32**	**4.24**	**65.09**	**27.34**

4.1 Synthetic Data Generation

We began by selecting passages from two in-domain documents, chosen by a domain specialist: the *ESA Space Environment Report* and the *ZERO DEBRIS CHART: Towards a Safe and Sustainable Space Environment*. Following the approach in [21], and to provide high-quality questions for synthetic data generation, the specialist crafted one or more questions for each passage aligned with five distinct task types: **Direct Q&A**, **Abstract Q&A**, **Fact Checking**, **Summarization**, and **Analysis**. Each task represents common real-world use cases of the pipeline according to our domain expert and tests different capabilities of the system.

- **Direct Q&A**: Questions with answers that are explicitly stated in the text.
- **Abstract Q&A**: Questions that require synthesizing or paraphrasing content rather than extracting it verbatim.
- **Fact Checking**: The model must determine whether a given statement is supported, refuted, or unverifiable based on the passage.
- **Summarization**: Requires the model to generate a concise summary of the chunk.
- **Analysis**: Involves higher-order thinking, such as drawing implications or making evaluative judgments based on the content.

We gathered 62 question–passage pairs, which we refer to as our *golden samples*. These high-quality, human-annotated examples served as the foundation for generating additional data across the rest of the corpus.

For each chunk and each task, we sampled five representative question–passage pairs from the golden samples. Following [21], we included these examples in the input prompt to the LLM to leverage its few-shot learning capabilities, which have been shown to improve the quality of synthetic data generation [22]. The model used for this generation was `Llama 3.3 70B` [23], running with 8-bit quantization. Using this setup, we generated 1,506 questions per task. We then did a filtering step, similar to [21], which has been shown to effectively remove low-quality synthetic queries [24, 25] and ensure the overall quality and relevance of the dataset. The details of this process are provided in App. B. In total, we obtained 7,036 questions, each generated from a semantically coherent passage of up to 2,000 tokens. These chunked passages also constitute the retrieval corpus used by the pipeline during evaluation.

5. Context Relevance Evaluation

We use **BM25** [14] and **Qwen2-1.5B** [2] [26] as our baseline retrievers, **BGE-M3**[17] as our reranker. Our goal is to evaluate the standalone performance of the retriever and to assess the additional impact of the reranker. In particular, we aim to answer the question: **"Is the reranking step actually necessary?"** In other words, for our specific task, does reranking significantly improve the quality of the retrieved passages, or are the retriever outputs alone sufficient? Since the drawbacks of reranking are non-negligible, we provide further discussion in App.A.

To answer this question, we ask the LLM itself to evaluate the relevance of each retrieved passage, both before and after reranking. For this purpose, we use **Llama 3.3 70B** as our evaluator. The use of LLMs for evaluation is already established in frameworks such as RAGAS [27] and is increasingly adopted in advanced retrieval pipelines [8], where the model acts as a judge. Recent adaptive retrieval approaches further demonstrate that LLMs can reliably decide which passages to retain or discard, as well as when to

[2] `https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct`

actively retrieve additional information in an agentic manner [8]. Employing an LLM as an automatic evaluator therefore allows us to directly measure whether reranking meaningfully improves the quality of the retrieved content.

We design a custom prompt (Fig.A1) that instructs the model to score each question–passage pair on a scale from 0 to 3, where 0 indicates complete irrelevance and 3 indicates high relevance. This fine-grained scoring scheme is motivated by prior work showing that multi-level relevance labels improve the performance of LLM-based classifiers [28]. We evaluate performance across four top-k ranges: 3, 5, 7, and 10. For reranking, we first retrieve the top-100 passages with the respective retriever and then rerank them to match the target top-k range.

As shown in Table 1, the reranked results consistently reduce the proportion of irrelevant passages (scores 0 and 1) across all top-k settings. For instance, at Top-3 with BM25, the reranker lowers the percentage of completely irrelevant passages from 2.12% to 1.29% while increasing the share of highly relevant ones (score 3) from 39.66% to 44.76%. This pattern holds across the evaluated top-k values, indicating that reranking effectively filters out less relevant passages and boosts the proportion of highly relevant results.

For moderately relevant passages (score 2), however, the retriever slightly outperforms the reranker in most settings. This suggests that reranking tends to trade some mid-level relevance for gains in the highest relevance category (score 3), while simultaneously reducing the frequency of irrelevant outputs (scores 0 and 1). The only exception occurs at Top-10, where the reranker achieves a marginal improvement in score 2. Notably, the Top-10 setting also exhibits the smallest differences overall, implying that reranking is most beneficial at lower top-k values, where ranking decisions carry greater weight.

Overall, both BM25 and Qwen2-1.5B demonstrate strong performance, with the majority of retrieved and reranked passages (scores 2 and 3) exceeding 90% relevance across most settings, except for the BM25 retriever at Top-10. Moreover, the LLM consistently shows a preference for the Qwen2-1.5B embedding model, both in its raw retrieval outputs and in the reranked passages, outperforming the BM25-based method.

6. Answer Quality Evaluation

In this section, we evaluate the answers generated by our pipeline under controlled retrieval conditions. To simulate both normal and adverse retrieval scenarios, we vary the number of unrelated passages included in the retrieved context. This approach allows us to mimic realistic situations where the retriever may return partially relevant or noisy information, which can occur due to query ambiguity, limited coverage of the knowledge base, or even deliberate malicious manipulation of content. By introducing these controlled perturbations, we can better understand the model's robustness and its vulnerability to exploitation. This setup allows us to test the system's robustness to two key challenges: (i) accidental retrieval errors from an underperforming retriever, and (ii) deliberate insertion of misleading passages through adversarial intervention. . For answer generation, we use **Llama 3 8B** [23], while evaluation is performed by the more capable **Llama 3.3 70B** [23], ensuring that a stronger judge model assesses the outputs of a mid-sized model that offers a practical balance between efficiency and performance.

Retrieval Settings. We consider three retrieval settings: (a) *Standard*, where the retriever provides only relevant passages (ideal case); (b) *3 Wrong*, where three unrelated in-domain passages are added to the retrieved set, creating a moderate level of noise; and (c) *5 Wrong*, where five unrelated in-domain passages are injected, representing a heavily corrupted retrieval scenario.

Evaluation Metrics. To measure system performance, we evaluate generated answers using four complementary metrics, each assessed with a carefully crafted prompt (see App. C): (1) *Answer Faithfulness*, which measures consistency with the retrieved context, ensuring no hallucinations or contradictions; (2) *Answer Relevance*, which captures how directly and effectively the response addresses the user's query; (3) *Noise Robustness*, which quantifies the ability to handle irrelevant or tangential information without degradation in quality; and (4) *Negative Rejection*, which reflects the model capacity to abstain from answering when the retrieved passages lack sufficient evidence.

Table 2: Evaluation of answer quality across different settings. **Standard** denotes clean retrieval, while **3 Wrong** and **5 Wrong** correspond to contexts with three or five unrelated passages injected.

Metric	Standard	3 Wrong	5 Wrong
Faithfulness	$4.42_{\pm 0.55}$	$4.41_{\pm 0.53}$	–
Relevance	$4.44_{\pm 0.60}$	$4.39_{\pm 0.60}$	–
Noise Robustness	–	$4.87_{\pm 0.68}$	–
Negative Rejection	–	–	$3.14_{\pm 1.55}$

The results show that in the *Standard* and *3 Wrong* conditions, answer faithfulness and relevance remain stable, indicating that the system tolerates moderate noise without

significant performance loss. The strong noise robustness score in the *3 Wrong* setting further supports this observation. However, under the *5 Wrong* condition, faithfulness, relevance and noise robustness can not be measured, and the system instead relied on negative rejection, with a mean score of 3.14. This suggests that when the context becomes dominated by irrelevant passages, the pipeline is more likely to abstain from providing an answer rather than risk hallucination.

Overall, these findings highlight the pipeline resilience to moderate retrieval corruption under the effect of state-of-the-art techniques for conditioning LLM outputs, in filtering noisy inputs.

7. Conclusions

In this work, we systematically evaluated the principal components of a RAG pipeline using state-of-the-art models. Our results show that current models are capable of supporting deployment in space operations; however, careful monitoring is essential, as even small modifications can have significant impacts on performance. Pipeline customization and task-specific adaptation are critical factors that must be considered when designing reliable systems for mission contexts. Due to the absence of standardized requirements for such pipelines and the broad scope of this study, we leave the detailed design of task-optimized pipelines for future work.

Acknowledgements

This research was carried out under Project "Artificial Intelligence Fights Space Debris" Nº C626449889-0046305 co-funded by Recovery and Resilience Plan and NextGeneration EU Funds (www.recuperarportugal.gov.pt), and by NOVA LINCS (UIDB/04516/2020) with the financial support of FCT.IP.

References

[1] Sobhana M., Syama Gudipati, and Satwik Panda. Llm based expert ai agent for mission operation managementekspert agenta ai z opcją llm do zarządzania operacjami misji. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, 15:88–94, 03 2025. doi: 10.35784/iapgos.6694.

[2] Clemens Schefels, Carsten Hartmann, Leonard Schlag, and Kathrin Helmsauer. Evaluating large language models for space operations. 09 2024.

[3] Andres Garcia-Silva, Cristian Berrio, Jose Manuel Gomez-Perez, Jose Antonio Martínez-Heras, Alessandro Donati, and Ilaria Roma. Spaceqa: Answering questions about the design of space missions and space craft concepts. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3306–3311, 2022.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[5] Audrey Berquand, Paul Darm, and Annalisa Riccardi. Spacetransformers: Language modeling for space systems. *IEEE Access*, 9:133111–133122, 2021. doi: 10.1109/ACCESS.2021.3115659.

[6] Paul Darm, Antonio Valerio Miceli Barone, Shay B. Cohen, and Annalisa Riccardi. DISCOSQA: A knowledge base question answering system for space debris based on program induction. In Sunayana Sitaram, Beata Beigman Klebanov, and Jason D Williams, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 487–499, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-industry.47. URL https://aclanthology.org/2023.acl-industry.47/.

[7] Matthew Foutter, Daniele Gammelli, Justin Kruger, Ethan Foss, Praneet Bhoj, Tommaso Guffanti, Simone D'Amico, and Marco Pavone. Space-llava: a vision-language model adapted to extraterrestrial applications. *arXiv preprint arXiv:2408.05924*, 2024.

[8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen

Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2:1, 2023.

[9] Ashok Urlana, Charaka Vinayak Kumar, Ajeet Kumar Singh, Bala Mallikarjunarao Garlapati, Srinivasa Rao Chalamala, and Rahul Mishra. Llms with industrial lens: Deciphering the challenges and prospects–a survey. *arXiv preprint arXiv:2402.14558*, 2024.

[10] Lizhou Fan, Lingyao Li, Zihui Ma, Sanggyu Lee, Huizi Yu, and Libby Hemphill. A bibliometric review of large language models research from 2017 to 2023. *ACM Transactions on Intelligent Systems and Technology*, 15(5):1–25, 2024.

[11] Weixin Liang, Yaohui Zhang, Mihai Codreanu, Jiayu Wang, Hancheng Cao, and James Zou. The widespread adoption of large language model-assisted writing across society. *arXiv preprint arXiv:2502.09747*, 2025.

[12] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. pages 15696–15707, 2023.

[13] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

[14] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3: 333–389, 01 2009. doi: 10.1561/1500000019.

[15] Priyanka Mandikal and Raymond Mooney. Sparse meets dense: A hybrid approach to enhance scientific document retrieval. In *The 4th CEUR Workshop on Scientific Document Understanding, AAAI*, 2024.

[16] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292, 2021.

[17] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

[18] Xinya Du and Heng Ji. Retrieval-augmented generative question answering for event argument extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4649–4666, 2022.

[19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[20] Isaac de Oliveira. semchunk: Segmenting documents with semantic awareness. https://github.com/isaacus-dev/semchunk, 2024. Accessed: 2025-06-13.

[21] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 338–354, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.20. URL https://aclanthology.org/2024.naacl-long.20/.

[22] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, 2023.

[23] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 Herd of Models. *arXiv e-prints*, pages arXiv–2407, 2024.

[24] Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Sultan, and Christopher Potts. Udapdr: Unsupervised domain adaptation via llm prompting and distillation of rerankers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11265–11279, 2023.

[25] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*, 2023.

[26] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive

learning. *arXiv preprint arXiv:2308.03281*, 2023.

[27] Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, 2024.

[28] Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. Beyond Yes and No: Improving Zero-Shot Pointwise LLM Rankers via Scoring Fine-Grained Relevance Labels. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024.

Appendix A. Reranker

Reranking serves a dual purpose: it prioritizes relevant passages chunks earlier in the prompt and filters out irrelevant ones, thereby enabling LLMs to extract key information more effectively. Typically, reranking employs a cross-encoder model, inspired by BERT [4], which evaluates the similarity between a query and a sentence during inference. Unlike retrievers, which compute similarity scores using cosine similarity between precomputed embeddings, rerankers allow all tokens in the query to interact with all tokens in the candidate sentence. This full cross-attention mechanism results in a richer similarity score and generally leads to higher performance.

Despite their effectiveness, rerankers cannot replace retrievers due to their computational complexity. Specifically, reranking has a complexity of $O(Q \times D)$, where Q is the number of queries and D is the number of passages, since each query must be compared to every passage at runtime. In contrast, retrievers have a complexity of $O(Q)$, as all passages embeddings are computed offline, allowing for efficient retrieval at scale.

Appendix B. Filter Step

We applied a filtering step to ensure the overall quality and relevance of the dataset, following a methodology similar to [21], which has been shown to effectively eliminate low-quality synthetic queries [24, 25]. If the original chunk for a question did not rank within the top 50 most similar chunks, we considered the question insufficiently grounded and removed it. This process ensured that only contextually meaningful and task-aligned questions were retained in the final dataset. For both this evaluation and the previous analysis, we used the `BGE-M3` model [17].

Appendix C. Prompts used for evaluation

System prompt:
You are an AI assistant that judges the relevance of a document to a given question. Respond with a score from 0 to 3.
QUESTION: User question
DOCUMENT: doc
Rate how relevant the document is to answering the question using the following scale:
0 = Completely irrelevant
1 = Slightly irrelevant
2 = Slightly relevant
3 = Completely relevant
Respond with a single number between 0 and 3.

Fig. A1: Document Relevance Prompt used to test both the Retriever and the Reranker.

System prompt:
You are a careful evaluator. Score the answer according to the rubric.
###Task Description:
A question, generated answer, retrieved documents, and a score rubric are given. The output format must be: "<result>[RESULT]</result>" (an integer between 1 and 5).No other explanations.
###Question: User question
###Retrieved Documents: Context
###Generated Answer: Generated Answer
###Generated Answer:
###Score Rubrics:
Answer Faithfulness ensures that the generated responses accurately reflect the information in the retrieved context, maintaining consistency and avoiding contradictions.
Score 1: Completely inaccurate, major contradictions.
Score 2: Mostly inaccurate, some contradictions.
Score 3: Somewhat accurate, minor inconsistencies.
Score 4: Largely accurate, no major issues.
Score 5: Fully faithful, perfectly reflects context.

Fig. A2: **Answer Faithfulness Prompt:** Used to evaluate the accuracy of generated answers against retrieved context.

System prompt:
You are a careful evaluator. Score the answer according to the rubric.
###Task Description:
A question, generated answer, retrieved documents, and a score rubric are given. The output format must be: "<result>[RESULT]</result>" (an integer between 1 and 5).No other explanations.
###Question: User question
###Retrieved Documents: Context
###Generated Answer: Generated Answer
###Generated Answer:
###Score Rubrics:
Answer Relevance measures how directly the generated responses address the given questions, ensuring they effectively respond to the core questions.
Score 1: Entirely irrelevant.
Score 2: Mostly irrelevant.
Score 3: Partially relevant.
Score 4: Mostly relevant.
Score 5: Fully relevant.

Fig. A3: **Answer Relevance Prompt:** Used to evaluate how well the generated answers address the question.

System prompt:
You are a careful evaluator. Score the answer according to the rubric.
###Task Description:
A question, generated answer, retrieved documents, and a score rubric are given. The output format must be: "<result>[RESULT]</result>" (an integer between 1 and 5).No other explanations.
###Question: User question
###Irrelevant Documents: Irrelevant Context
###Generated Answer: Generated Answer
###Generated Answer:
###Score Rubrics:
Noise Robustness evaluates whether the model's answer includes any information taken from irrelevant documents. This ensures the model can resist being influenced by distracting, unrelated context.
Score 1: The answer clearly uses or is based on the irrelevant documents.
Score 3: The answer is partially influenced or overlaps slightly with the irrelevant documents.
Score 5: The answer does not contain any information from the irrelevant documents.

Fig. A4: **Noise Robustness Prompt:** Used to evaluate the model's resistance to irrelevant information.

System prompt:
You are a careful evaluator. Score the answer according to the rubric.

###Task Description:

A question, generated answer, retrieved documents, and a score rubric are given. The output format must be: "<result>[RESULT]</result>" (an integer between 1 and 5).No other explanations.

###Question: User question

###Retrieved Documents: Context

###Generated Answer: Generated Answer

###Generated Answer:

###Score Rubrics:

Negative Rejection assesses the modelâ€™s ability to avoid generating a response when the retrieved documents do not contain the necessary information to answer the query.

Score 1: The model provides a confident or specific answer despite no relevant information being present in the documents.

Score 3: The model gives a vague or hedging answer that suggests uncertainty, but still attempts to answer without sufficient support.

Score 5: The model appropriately refrains from answering or clearly states that the answer cannot be found in the provided documents.

Fig. A5: **Negative Rejection Prompt:** Used to evaluate the model's ability to refrain from answering when the information is insufficient.