Retrieval-Augmented Generation (RAG) To Enhance Open Testbed Documentation

Saieda Ali Zada
saieda@udel.edu
University of Delaware
Newark, DE, US

Marc Richardson [advisor]
mtrichardson@uchicago.edu
University of Chicago
Chicago, IL, US

Kate Keahey [advisor]
keahey@uchicago.edu
Argonne National Laboratory
Lemont, IL, US

ABSTRACT

Researchers in high-performance computing (HPC) and cloud environments encounter disparate sources of documentation and difficulties finding accurate information. This can cause inefficiency, increase the reliance on support teams, and change the focus of the researcher from the main the experiment. To address these challenges, we developed an AI powered search system leveraging large language models (LLMs) with Retrieval-Augmented Generation (RAG) to unify various documentation sources and provide accurate, context-aware answers with cited references to relevant sources. We evaluated our RAG system with Chameleon Cloud testbed documentation as a case study, finding that our RAG system outperforms other generic LLMs in answering a variety of user questions and performs comparable to proprietary LLMs when properly tuned and optimized.

KEYWORDS

High-performance Computing, Chameleon Cloud, Retrieval Augmentation, Large-Language Models

ACM Reference Format:
Saieda Ali Zada, Marc Richardson [advisor], and Kate Keahey [advisor]. 2025. Retrieval-Augmented Generation (RAG) To Enhance Open Testbed Documentation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Large-scale scientific infrastructure, such as a research testbed, is crucial for advancing science by enabling complex, large-scale experiments in CS systems research. Despite the availability of documentation for these systems, users often find it challenging to navigate the extensive materials, leading to missing or inconsistent information. This inefficiency slows down research and redirects support staff from other critical duties to user assistance.

Recent advancements in Large Language Models (LLMs), particularly Retrieval-Augmented Generation (RAG) [12], offer a new solution to this knowledge gap. RAG-enabled LLMs can generate accurate answers with specific references, providing immediate, high-quality assistance to users and freeing up engineering time. This poster details the development and evaluation of a RAG system using Chameleon Cloud [11] testbed documentation as a case study. Our RAG system demonstrates superior performance compared to other generic LLMs in answering various user questions about Chameleon and performs comparably to proprietary LLMs when properly optimized.

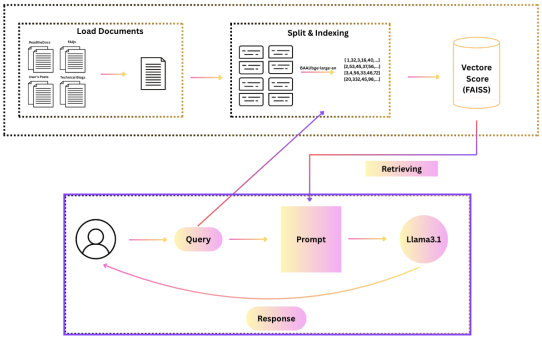

Figure 1: Architecture diagram of our RAG system

2 APPROACH

Our approach employs a standard RAG architecture [12] built with the LangChain framework [14]. Figure 1 illustrates the three main components.

Document Processing: We collected sources collected from the main Chameleon ReadtheDocs website [5], FAQs [7], blogs [4], and public posts from the Forum [6]. We retrieved the HTML page for each source, formatted it to remove low-value content (e.g., website headers and footers), and recoreded its URL.

Indexing: We divided documents into "chunks" – short text segments (500 to 2000 tokens or words) to enhance retrieval precision and ensure compatibility with LLM context windows [18]. Each chunk is converted into a numerical vector representation using the "BAAI/bge-large-en" embedding model [3], allowing the system to compare semantic similarity rather than just keywords. These embeddings are stored in a FAISS [10] vector database for quick and efficient similarity searches.

Retrieval and Generation: When a user submits a query (e.g., "How do I access Chameleon hardware?"), the system retrieves the most relevant sources from the vector store. These sources, along with the original query, are then passed to an LLM model (specifically, the open-source "Meta Llama 3.1" [15]). A system prompt instructs the LLM to generate an accurate answer using the

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

provided context. The system prompts used for different models are documented at [1].

Experimental Setup: The work was conducted across various hardware environments, including the Perlmutter supercomputer for initial implementation, an Apple M1 system for early development, and a KVM instance on Chameleon with 1 NVIDIA H100 GPU (40GB) for deployment and accelerated evaluation. GPU acceleration significantly reduced the latency from user query submission to answer generation from 10 minutes to 20 seconds.

3 EVALUATION METHODOLOGY

For evaluation, we used 20 test queries on Chameleon features and policies with ground-truth answers that were generated by Gemini 2.5 Pro [9] and revised by a Chameleon engineer. We compared our model answers against these ground-truth answers, as well as a positive baseline (answers from OpenAI's GPT-5 [16], expected to perform best) and a negative baseline (answers from Llama 3.1 [15] without RAG, expected to perform worst).

Fourteen configurations of our RAG model (Models 1-14) were evaluated by varying parameters such as documentation sources, chunking sizes, number of matches returned, prompt techniques, re-ranking, and a Dual-LLM design (full model specifications are available at [1]).

We employed two primary evaluation methods to compare models and baselines with the ground truth.

3.1 Similarity Metrics

We measured the statistical similarity between our generated answers and the ground-truth answer using Cosine similarity (vector similarity) [17], Jaccard score (set similarity) [17], ROUGE-L (longest common subsequence) [13], and BERTScore (contextual embeddings) [19]. These metrics are presented in Table 1 and utilize different algorithms to quantify similarity in unstructured text data.

3.2 Judge Model

We used Anthropic's "Claude 3.5 Sonnet" [2] as a judge to evaluate and compare answer quality. In each comparison, the judge model was presented with one answer from a baseline and one from a RAG model for the query. It evaluated, scored, and compared both answers against the ground-truth answer, selecting a winning answer or declaring a tie if both were of similar quality. This methodology is often more effective than similarity metrics at detecting quality differences across LLMs for specific tasks like information retrieval [20], sometimes even matching human quality assessment [8].

4 RESULTS AND DISCUSSION

Aggregate results from our similarity metrics for RAG and baseline models are presented in Table 1. Our findings indicate that these standard similarity metrics were insufficient for detecting differences in the overall quality of generated answers, confirming previous research [8, 20]. Almost all our models achieved similar performance across these metrics, even with changes to critical parameters like documentation sources or system prompts.

The judge model evaluation provided clear insights. Tables 2 and 3 show that early models (1-6) with limited sources performed poorly. However, later models (7-14) improved significantly by adding more documentation, enhancing the system prompt, and using a two-stage similarity search. These models consistently outperformed the negative baseline and achieved comparable performance to the positive baseline (e.g., Model 12).

5 CONCLUSION

In this work, we developed and studied a RAG-based search system to improve access to Chameleon Cloud documentation. Our findings highlight both the potential and limitations of RAG for scientific documentation. While not a complete replacement for leading proprietary models, a well-designed RAG system provides clear benefits over generic models and potentially proprietary models with sufficient optimization. Future work will focus on integrating specialized sources like user ticket data with privacy safeguards, refining a Dual-LLM architecture for user output generation, and exploring more effective user-feedback-based evaluation metrics.

ACKNOWLEDGMENTS

The results of this poster were obtained on the Chameleon Testbed funded by the National Science Foundation (Award No. 2431425). We are grateful to Kexin Pei at the University of Chicago for his helpful comments.

REFERENCES

[1] Ali Zada, Saieda and Richardson, Marc. RAG-docs-chameleon. https://github.com/UD-CRPL/RAG-docs-chameleon, 2024. Accessed: 2025-08-25.
[2] Anthropic. Introducing Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet, 2025. Accessed: 2025-08-17.
[3] Beijing Academy of Artificial Intelligence (BAAI). BAAI bge-large-en Model. https://huggingface.co/BAAI/bge-large-en, 2025. Accessed: 2025-08-17.
[4] Chameleon Cloud. Chameleon cloud blog. https://www.chameleoncloud.org/blog. Accessed: 2025-08-25.
[5] Chameleon Cloud. Chameleon documentation. https://chameleoncloud.readthedocs.io/en/latest/. Accessed: 2025-08-25.
[6] Chameleon Cloud. Chameleon users forum. https://forum.chameleoncloud.org/. Accessed: 2025-08-25.
[7] Chameleon Cloud. Frequently asked questions. https://www.chameleoncloud.org/help/faq. Accessed: 2025-08-25.
[8] Chiang, T.-H., Hsieh, C.-W., Chuang, Y.-S., Lo, C.-Y., and Lee, H.-y. Can LLMs be trusted for evaluating RAG systems? a survey of methods and datasets. *arXiv preprint arXiv:2407.11181* (2024).
[9] Google DeepMind. Gemini 1.5 Pro. https://deepmind.google/models/gemini/pro/, 2025. Accessed: 2025-08-17.
[10] Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data 7*, 3 (2019), 535–547.
[11] Keahey, K., Anderson, J., Zhen, Z., Riteau, P., Ruth, P., Stanzione, D., Cevik, M., Colleran, J., Gunawi, H. S., Hammock, C., Mambretti, J., Barnes, A., and Halstead, F. Lessons learned from the chameleon testbed. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)* (2020), USENIX Association, pp. 219–233. website:https://chameleoncloud.org/.
[12] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Ott, M., Chen, W.-t., Conneau, A., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems* (2020), vol. 33, pp. 9459–9474.
[13] Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out* (2004).
[14] Mavroudis, V. Langchain. *arXiv preprint arXiv:2405.08933* (2024).
[15] Meta AI. Introducing Meta LLaMA 3.1. https://ai.meta.com/blog/meta-llama-3-1/, 2025. Accessed: 2025-08-17.
[16] OpenAI. Introducing GPT-5. https://openai.com/index/introducing-gpt-5/, 2025. Accessed: 2025-08-17.
[17] Rinjeni, T. P., Indriawan, A., and Rakhmawati, N. A. Matching scientific article titles using cosine similarity and jaccard similarity algorithm. *Procedia Computer Science 234* (2024), 553–560.
[18] Salton, G., Wong, A., and Yang, C. S. A vector space model for automatic indexing. *Communications of the ACM 18*, 11 (1975), 613–620.
[19] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning*

Model	ROUGE-score			BERT-score			Jaccard-score			Cosine-score		
	Ave	Max	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max	Min
model_1_answer	0.1942	0.3980	0.1206	0.7784	0.8322	0.7342	0.1287	0.2875	0.0423	0.4083	0.5975	0.1737
model_2_answer	0.2041	0.3980	0.1141	0.7758	0.8280	0.7409	0.1319	0.3793	0.0411	0.4026	0.6116	0.1996
model_3_answer	0.2044	0.4762	0.1290	0.8397	0.8397	0.7234	0.1522	0.5663	0.0270	0.4230	0.5861	0.1721
model_4_answer	0.2037	0.3318	0.1474	0.7813	0.8462	0.7359	0.1507	0.2532	0.0253	0.4279	0.5968	0.1488
model_5_answer	0.2036	0.4762	0.0942	0.7758	0.8397	0.7097	0.1502	0.5663	0.0270	0.4269	0.6904	0.2221
model_6_answer	0.2081	0.3459	0.1359	0.7761	0.8353	0.7296	0.1320	0.3438	0.0267	0.4060	0.6194	0.1869
model_7_answer	0.2087	0.3693	0.1156	0.7897	0.8296	0.7195	0.1625	0.4579	0.0678	0.4991	0.7591	0.1625
model_8_answer	0.2158	0.3892	0.1329	0.7950	0.8458	0.7616	0.1680	0.4182	0.0769	0.4997	0.7403	0.1367
model_9_answer	0.2010	0.2667	0.1496	0.7874	0.8332	0.7468	0.1419	0.2475	0.0732	0.4551	0.6774	0.2222
model_10_answer	0.1974	0.3243	0.1359	0.7878	0.8262	0.7618	0.1503	0.2410	0.0763	0.4657	0.7604	0.2153
model_11_answer	0.2157	0.4244	0.1399	0.7900	0.8547	0.7268	0.1743	0.3486	0.0531	0.4621	0.6683	0.1897
model_12_answer	0.1992	0.3832	0.1388	0.7695	0.8610	0.6741	0.1841	0.5862	0.0548	0.4435	0.6805	0.1419
model_13_answer	0.1884	0.2989	0.0360	0.7681	0.8111	0.6360	0.1833	0.2286	0.0185	0.4429	0.6453	0.0475
model_14_answer	0.2061	0.3553	0.1442	0.7796	0.8370	0.7027	0.1815	0.6364	0.0680	0.4703	0.7587	0.2091
base_openai_model	0.2031	0.3503	0.1148	0.8020	0.8528	0.7546	0.1807	0.2969	0.1087	0.4632	0.6829	0.2606
base_ollama_model	0.2052	0.3265	0.1081	0.8003	0.8379	0.7361	0.1446	0.2632	0.0505	0.4457	0.7107	0.1336

Table 1: Evaluation similarity metrics (ROUGE-score, BERT-score, Jaccard-score, Cosine-score) with average, max, and min values for each model and baseline.

Models vs. Negative Baseline	Loss	Tie	Win
model-1 answer	12	1	7
model-2 answer	8	2	10
model-3 answer	6	2	12
model-4 answer	6	1	13
model-5 answer	5	2	13
model-6 answer	7	1	12
model-7 answer	0	1	19
model-8 answer	1	0	19
model-9 answer	2	0	18
model-10 answer	4	1	15
model-11 answer	0	1	19
model-12 answer	0	1	19
model-13 answer	1	0	19
model-14 answer	0	2	18

Table 3: Comparison outcomes between RAG models answers and negative baseline answers (Llama 3.1).

Models vs. Positive Baseline	Loss	Tie	Win
model-1 answer	20	0	0
model-2 answer	19	0	1
model-3 answer	19	0	1
model-4 answer	18	0	2
model-5 answer	19	0	1
model-6 answer	19	0	1
model-7 answer	13	0	7
model-8 answer	14	1	5
model-9 answer	18	0	2
model-10 answer	13	1	6
model-11 answer	14	1	5
model-12 answer	11	0	9
model-13 answer	15	0	5
model-14 answer	13	1	6

Table 2: Comparison outcomes between RAG models answers and positive baseline answers (OpenAI's GPT-5).

Representations (ICLR 2020) (2019).

[20] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685* (2023).

Received 25 August 2025