

APPLICATION OF LARGE LANGUAGE MODELS FOR THE EXTRACTION OF INFORMATION FROM PARTICLE ACCELERATOR TECHNICAL DOCUMENTATION

Q. Dai*, R. Ischebeck, M. Sapinski,
PSI Center for Accelerator Science and Engineering, Villigen, Switzerland
A. Grynier†, Google DeepMind, Zurich

Abstract

The large set of technical documentation of legacy accelerator systems, coupled with the retirement of experienced personnel, underscores the urgent need for efficient methods to preserve and transfer specialized knowledge. This paper explores the application of large language models (LLMs), to automate and enhance the extraction of information from particle accelerator technical documents. By exploiting LLMs, we aim to address the challenges of knowledge retention, enabling the retrieval of domain expertise embedded in legacy documentation. We present initial results of adapting LLMs to this specialized domain. Our evaluation demonstrates the effectiveness of LLMs in extracting, summarizing, and organizing knowledge, significantly reducing the risk of losing valuable insights as personnel retire. Furthermore, we discuss the limitations of current LLMs, such as interpretability and handling of rare domain-specific terms, and propose strategies for improvement. This work highlights the potential of LLMs to play a pivotal role in preserving institutional knowledge and ensuring continuity in highly specialized fields.

INTRODUCTION

Having a well-structured and complete documentation is crucial for maintaining and developing any large industrial system. In case of accelerator facilities, such as the High Intensity Proton Accelerator (HIPA) [1] at PSI, the available documentation is often sparse and inaccurate. HIPA has been designed and build 50 years ago, before any electronic documentation was invented. Furthermore, as an experimental facility, it evolved significantly over years. The absence of a consistent long-term documentation strategy has further contributed to the current state of confusion.

Many accelerators facilities face similar problems. Newer machines usually have more consistent documentation. For instance Proscan, which is a proton therapy facility in PSI, is well documented as it was required by licensing authorities. Also any change done to medical facility must be carefully analyzed and approved before implementation. As a result Proscan, which is 20 years old, is almost immutable with respect to HIPA.

The expertise is not only in bare set of documentation. When a new specialist takes over the responsibility, he or she needs a certain amount of time, counted in years, to

reach the same level of performance as the retiring expert. Simulating the ability to chat (questions and answers) with an expert can be a great tool to speed up this process.

In this work we took beam instrumentation of HIPA and Proscan as an example to develop LLM-based chatbot which could help, based on existing documentation, to work on technical issues of the facility. Similarly to other labs [2, 3], we choose to develop a system which runs locally, not exposing internal documentation.

THE DOCUMENTS

The initial set of documents included 58 pdf files, in English (36) and in German (22), written mostly by one physicist over the span of 30 years. Those documents are technical and specific to HIPA and Proscan. Before using them, they were checked if they are up to date. In addition a few other documents were included:

- Two master theses, containing general descriptions of HIPA and results of particular investigations.
- Several conference proceedings which include overview of HIPA and Proscan instrumentation [4] or more detailed analysis of problems related to particular devices, for instance see Ref. [5].
- Publicly available books with courses on beam instrumentation, e.g., Ref. [6].

During the tests it was found that an important part of the information - for instance the naming schemes or the location of electronics racks - is missing. Large part of these information are present in excel tables, schematics and databases, which we cannot process yet. The gap was partially filled by creating dedicated text files with the missing information.

The input documents are stored in *Corpus* directory and a script to process them is provided. In total they contain 1032 pages in English and 239 pages in German.

METHODS

Retrieval-augmented generation (RAG) combines the broad language ability of large language models (LLMs) with an external knowledge base, improving factual accuracy and grounding [7]. All data were processed offline, and every model was run locally with **Ollama** [8] on a Mac Studio M2 Ultra (192 GB unified memory).

Pipeline. Figure 1 shows the two-stage RAG workflow:

* qing.dai@psi.ch

† Contributed in an advisory capacity.

1. **Pre-processing.** PDFs are parsed with the open-source MINERU extractor [9]. We retain text, equations, and tables, then split each document into chunks and store (chunk \rightarrow embedding, file-id) tuples in a vector database. Embeddings are pre-computed for fast retrieval.

2. Runtime.

- Retrieval.** A user query is embedded with the same multilingual BGE-M3 model [10]; the top- k similar chunks are returned.
- Generation.** The query plus retrieved chunks are fed to the instruction-tuned gemma3:27b-it-fp16 [11] LLM, which produces the final answer.
- Evaluation.** Retrieval is scored with recall@ k and MRR@ k [12]; generation is scored with answer accuracy given a reference answer and mean confidence over correctly judged answers, using the same Gemma model as a judge.

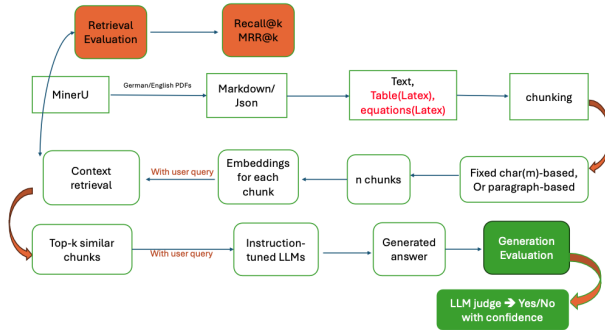


Figure 1: End-to-end RAG pipeline.

Chatbot interface. Figure 2 illustrate the front-end. Besides the answer, the bot lists the five most relevant files with similarity scores and snippets; clicking a file opens the exact location in the context.

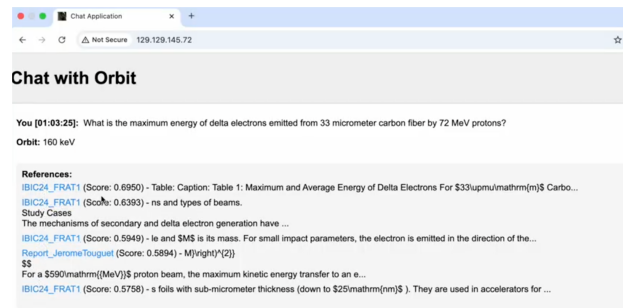
Benchmark. Two domain experts created 100 question-answer (QA) pairs, each linked to a gold reference file (70 English, 30 German). We use this set for both retrieval and generation evaluation.

RESULTS

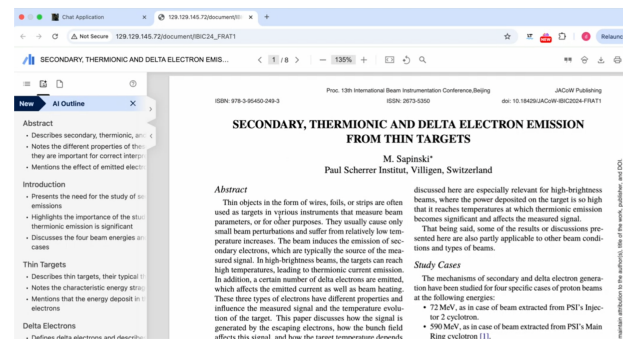
Chunking Strategies

We tested four splitting schemes:

- character windows of 800, 1600, and 2000 chars;
- paragraph windows (minimum 120 tokens, with short paragraphs merged).
- paragraph with context (previous and following paragraph): this scheme is used in generation, hoping to provide more context from offset, due to semantics which may be broken by chunking.



(a) Query and LLM answer.



(b) Clickable document reference.

Figure 2: Chatbot interface.

Retrieval Performance

Figure 3 summarizes recall@ k and MRR@ k for $k \in \{3, 5\}$ across all chunk sizes. Key observations:

- Top-5 > Top-3 for recall**, but MRR grows more modestly, reflecting the trade-off between depth and ranking quality.
- Smaller chunks outperform larger ones.** Neither 1600- nor 2000-char windows improved recall or MRR.
- Paragraph splitting offered no clear gain**, despite providing semantically complete units.
- German queries lagged behind English.** Translating German chunks to English—using the 4-bit gemma2:27b-instruct-q4_K_M [13] model—significantly boosted recall and MRR for German queries and lifted English queries, likely by reducing multilingual noise.

Generation Performance

Guided by retrieval results, we tested generation on the best two chunk sizes (800 and 1600 chars) and paragraph schemes with $k \in \{3, 5\}$ and three prompt variants: *no-translation* (k-N), *translation* (k-T) and *translation + chunk-score* (k-S) (the similarity score is appended and the LLM is instructed to focus on high-score chunks). Figure 4 reports answer accuracy and mean confidence. Key findings are:

- Translation helps at $k=3$ but not consistently at $k=5$.** Noise from additional chunks partly offsets the translation gain.

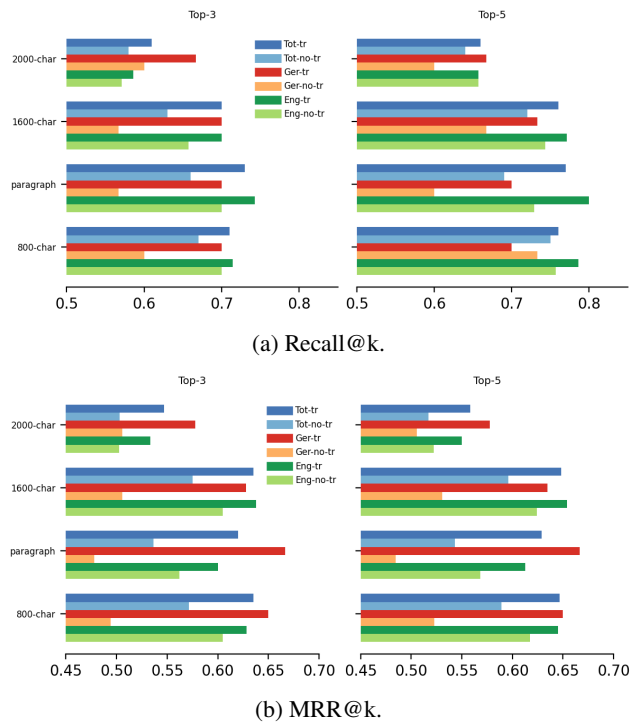


Figure 3: Retrieval comparison across chunking schemes before and after translation.

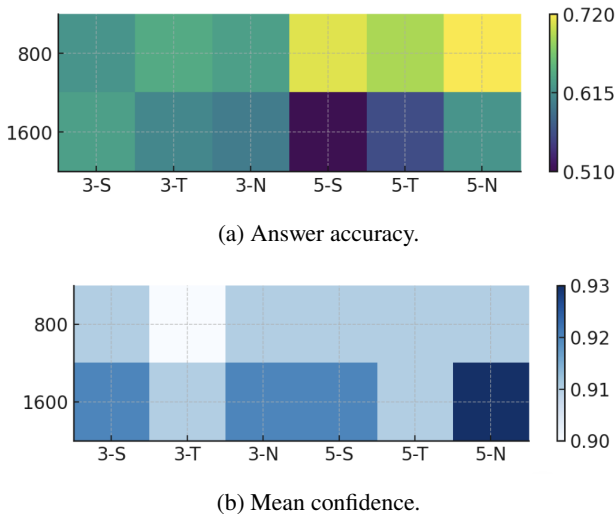


Figure 4: Generation results for 800- and 1600-char chunks under three prompt settings.

- Overall, the model judges the generated answer with high confidence, with a narrow range from 0.90 to 0.93.
- While Top-5 retrieval with 1600-character chunks matches the recall of 800-character chunks, it yields lower answer accuracy. Inspection of the misclassified outputs shows that only this setup suffers from hallucinations—either summarizing retrieved text out of context or ignoring the query altogether—errors absent in the other three chunk- k combinations. We traced the issue to Ollama’s default context window of 2048 tokens [14]. Although our Top-5

1600-char inputs average around 1500 tokens, they approach the limit, leading to truncation and hallucinations.

Paragraph-Level Chunking and Context Window

To test whether larger semantic units help generation, we evaluated two paragraph-based schemes at $k=3$ and 5:

- *Paragraph*: split on natural paragraph boundaries (min. 120 tokens).
- *Paragraph + Context*: each paragraph plus its immediate predecessor and successor.

To prevent the truncation-induced hallucinations we observed with 1600-char, Top-5 inputs, we increased Ollama context window to 6000 tokens for these experiments. Figure 5 shows answer accuracy for all paragraph variants. Adding context does not improve — and even slightly degrades — the accuracy, especially at $k=5$, confirming that simple paragraph splits suffice.

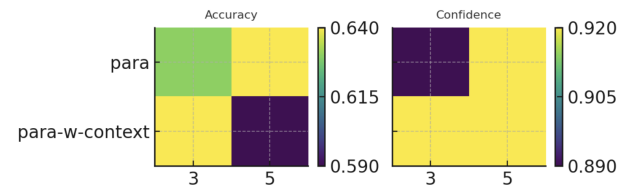


Figure 5: Answer accuracy for paragraph-only vs. paragraph-with-context at $k=3, 5$.

Recommendation. For our accelerator documentation, *800-char chunks with Top-5 retrieval* delivers the highest answer accuracy and confidence, illustrating the critical role of chunk size, Top- k , and prompt structure.

PERSPECTIVES

Initial evaluations demonstrate strong performance utilizing a RAG pipeline for question answering based on textual document content. However, a significant limitation remains in effectively retrieving information embedded within non-textual elements of the source documents, specifically tables and figures encompassing schematics, plots, and photos.

Current RAG implementations exhibit difficulty in retrieving and reasoning with visual information. Future work will focus on addressing this challenge through two paths: first, exploring pre-processing techniques such as automatic figure captioning to generate descriptive text prior to indexing, thereby enabling textual retrieval of visual content using the same RAG system that we are presently using; and second, the evaluation of multi-modal embedding models that would directly encode figure content into the vector database.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Rudolf Doelling, Marco Bocchio and Yingqiang Gao.

REFERENCES

- [1] M. Seidel *et al.*, “Production of a 1.3 MW proton beam at PSI,” in *Proc. IPAC’10*, Kyoto, Japan, paper TUYRA03, pp. 1309–1313, <https://accelconf.web.cern.ch/IPAC10/papers/tuyra03.pdf>
- [2] F. Mayet, “GAIA: A general AI assistant for intelligent accelerator operations,” *arXiv:2405.01359*, May 2024. doi:10.48550/arXiv.2405.01359
- [3] A. Sulc *et al.*, “Towards unlocking insights from logbooks using AI,” in *Proc. 15th Int. Part. Accel. Conf. (IPAC’24)*, Nashville, TN, USA, May 2024, pp. 3579–3582. doi:10.18429/JACoW-IPAC2024-THPR37
- [4] R. Dolling, “Diagnostics of the PROSCAN proton-therapy beam lines,” in *Proc. 6th European Workshop on Beam Diagnostics and Instrumentation for Particle Accelerators (DIPAC’03)*, Mainz, Germany, May 2023, pp. 152–154, <https://accelconf.web.cern.ch/d03/papers/PM25.pdf>
- [5] M. Sapinski, R. Dölling, and M. Rohrer, “Commissioning of the renewed long radial probe in PSI ring cyclotron,” in *Proc. 11th Int. Beam Instrum. Conf. (IBIC’22)*, Kraków, Poland, Sep. 2022, pp. 76–79. doi:10.18429/JACoW-IBIC2022-MOP19
- [6] P. Forck, “JUAS Lecture Notes on Beam Instrumentation and Diagnostics,” presented at the Joint University Accelerator School (JUAS’11), Archamps, France, Jan.-Mar. 2011, https://www-bd.gsi.de/conf/juas/juas_script.pdf
- [7] Y. Gao *et al.*, “Retrieval-augmented generation for large language models: A survey,” *arXiv:2312.10997*, Dec. 2023. doi:10.48550/arXiv.2312.10997
- [8] Ollama, <https://ollama.com/>
- [9] B. Wang *et al.*, “MinerU: An open-source solution for precise document content extraction,” *arXiv:2409.18839*, Sep. 2024. doi:10.48550/arXiv.2409.18839
- [10] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, “BGE M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation,” *arXiv:2402.03216*, Sep. 2024. doi:10.48550/arXiv.2402.03216
- [11] Gemma Team, Google DeepMind, *Gemma 3 Technical Report*, Mar. 2025, <https://google/Gemma3Report>
- [12] Pinecone, “Evaluation measures in information retrieval”, <https://www.pinecone.io/learn/offline-evaluation/>
- [13] Gemma Team, “Gemma 2,” Kaggle, 2024. doi:10.34740/KAGGLE/M/3301
- [14] Questions about context size No. 2204, Ollama, <https://github.com/ollama/ollama/issues/2204>