ApertusRAG - RAG Architectures for Apertus

Amine Louah, | 343896 | amine.louah@epfl.ch
Szabina Horvath Mikulas | 226459 | szabina.horvath-mikulas@epfl.ch
Rayane Zouagui, | 356531 | rayane.zouagui@epfl.ch

Abstract — We present an end-to-end system for retrieval-augmented generation (RAG) over large-scale web archives. Using the ETH Zürich Web Archive, which consists of thousands of temporally ordered web crawls and heterogeneous content, we address challenges in scalable text extraction, temporal redundancy, and retrieval over evolving documents. Our pipeline integrates archival data processing, time-aware deduplication, semantic indexing, and retrieval-augmented question answering, and is deployed on the CSCS high-performance computing infrastructure. While initial experiments validate the approach on representative data subsets, scaling to the full archive reveals practical challenges related to indexing stability and retrieval quality. We document these challenges and outline ongoing evaluation efforts for both base language models and retrieval-augmented configurations.

1 Introduction

Retrieval-augmented generation (RAG) enables language models to answer questions using external document collections, improving factual grounding and domain coverage (Lewis et al., 2020). Applying RAG to large-scale, temporally structured web archives introduces distinct challenges: retrieval systems must handle vast data volumes while avoiding irrelevant or outdated content, and generation must remain grounded in historically appropriate evidence.

Institutional web archives differ substantially from curated text corpora. They contain repeated crawls of the same URLs, significant boilerplate content, and heterogeneous document types. Without explicit processing, these properties degrade retrieval effectiveness and bias downstream generation.

This work is part of APERTUS, an open and transparent multilingual large language model initiative. We present an end-to-end system for question answering over the ETH Zürich Web Archive that integrates archival data processing, indexing, and retrieval-augmented inference. The primary focus of this paper is the data-centric processing pipeline and the practical challenges encountered when preparing archival web data for machine learning applications. Evaluation focuses on characterizing the impact of retrieval augmentation under realistic system constraints rather than on optimizing model performance.

2 System Implementation and Deployment

We focus on the concrete system delivered and deployed for large-scale archival retrieval on high-performance computing (HPC) infrastructure. The complete system is packaged as a containerized application and executed on the CSCS cluster using batch-based job scheduling.

Each pipeline stage - archive extraction, temporal deduplication, semantic indexing, retrieval, and retrieval-augmented generation - is implemented as an independent module within a single Docker container. This design enables reproducible execution across heterogeneous environments and avoids reliance on long-lived services, which are often unavailable on HPC systems.

Indexing is performed via stateless batch jobs that stream input data and write directly to a shared Elasticsearch backend. To ensure fault tolerance under job preemption and partial failures, document identifiers are derived deterministically from stable source attributes, making indexing idempotent and allowing safe re-execution without duplication.

The system supports parallel execution by logically partitioning the corpus using metadata-based filters (e.g., crawl year or document modality) rather than filesystem sharding. This approach enables scalable ingestion across multiple batch jobs while avoiding data duplication or complex file management.

During inference, the system operates in two configurations: a baseline language model mode and a retrieval-augmented mode. In the latter, a dense embedding model retrieves relevant archival content from the index, which is then incorporated into the prompt of a generative language model. All components are instantiated within the same containerized environment to ensure consistent dependencies and execution semantics.

2.1 Local-to-Cluster Development Path

Development proceeded incrementally from a local environment to distributed deployment on CSCS infrastructure. Initial experiments used a local Elasticsearch instance, locally hosted embedding models, and small representative data subsets, enabling rapid prototyping of the RAG pipeline.

Migration to the cluster environment exposed scaling and compatibility issues not present locally. In particular, the CSCS-hosted embedding service (`Snowflake/snowflake-arctic-embed-l-v2.0`) exposed an OpenAI-compatible interface that differed in subtle but critical ways from standard APIs, requiring a custom embedding wrapper for correct integration. Additionally, large-scale indexing against the remote Elasticsearch backend repeatedly failed due to oversized bulk requests (HTTP 413 errors), despite extensive configuration attempts.

As a result, the indexing pipeline was reimplemented with explicit control over batching behavior, request size, and document identifiers. Stabilizing ingestion and retrieval at scale required substantial engineering effort, which constrained opportunities for extensive hyperparameter tuning and large-scale evaluation. Consequently, evaluation emphasizes comparative behavior under fixed system configurations.

3 End-to-End System Overview

Figure 1 illustrates the end-to-end pipeline. Web archive data stored in WARC format is processed to extract text from HTML and PDF resources, normalized, temporally annotated, and deduplicated. The resulting corpus is indexed for semantic retrieval. At inference time, retrieved documents are incorporated into the model prompt to generate grounded answers.

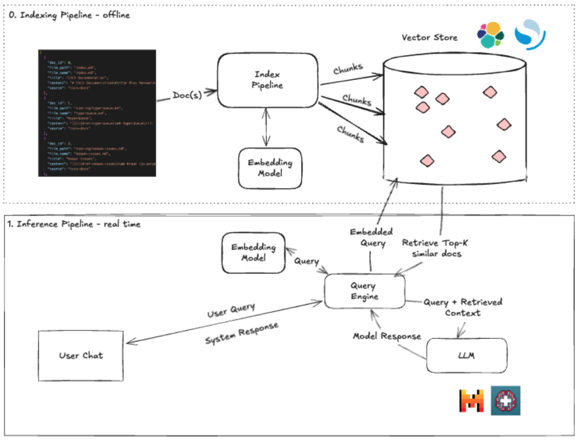

Figure 1: RAG architecture for APERTUS.

3.1 Source Data

We use the ETH Zürich Web Archive hosted on CSCS infrastructure, consisting of 12,887 compressed WARC files from the Archive-It collection *ETHZ Websites (2022–2025)*. Each WARC contains HTTP request–response records with metadata including URL, content type, HTTP status code, and capture timestamp.

3.2 Extraction Pipeline

A streaming extraction pipeline processes WARC files sequentially. For each record, we retain valid HTTP responses (status code 200) from approved ETH domains with HTML or PDF content types. HTML content is parsed and converted to plain text by removing markup and scripts. PDF content is extracted using a text-based parser, with unreadable files skipped. Extracted records are written incrementally to JSONL format, reducing the archive from several terabytes to approximately 15 GB of structured text.

3.3 Temporal Deduplication

To mitigate temporal redundancy while preserving historical variation, we apply time-aware deduplication that retains one snapshot per normalized URL per year. Records are grouped by

$$(\texttt{normalized\_url}, \texttt{year}(\texttt{capture\_time}))$$

and only the most recent record per group is retained. URL normalization removes superficial variations such as trailing slashes and common tracking parameters. After deduplication, the corpus is reduced to approximately 8 GB.

These preprocessing choices directly affect downstream retrieval behavior, particularly with respect to boilerplate prevalence, temporal relevance, and language distribution, which are revisited in the evaluation and error analysis.

4 Indexing and Retrieval

Documents are split into overlapping, sentence-aware chunks to improve retrieval granularity. Each chunk is embedded using a transformer-based dense embedding model. In the cluster deployment, embeddings are generated using `Snowflake/snowflake-arctic-embed-l-v2.0`, accessed via a custom wrapper to ensure compatibility with the surrounding pipeline.

Embedded chunks are indexed into a vector-enabled Elasticsearch index, storing the text, inherited metadata, and embedding vectors. Initial attempts using off-the-shelf ingestion frameworks failed at scale due to oversized bulk requests. A custom ingestion pipeline with explicit batching and payload size limits was therefore implemented, trading throughput for stability.

5 Evaluation

Our evaluation aims to assess the impact of retrieval augmentation on institutional question answering rather than to establish a general-purpose benchmark. Due to the substantial engineering effort required to stabilize large-scale indexing and retrieval on HPC infrastructure, evaluation focuses on a fixed set of ETH-specific queries and emphasizes comparative analysis between baseline and retrieval-augmented configurations.

5.1 Baseline vs RAG Performance

We evaluate two models (Apertus-8B and Qwen3-8B) in both baseline (no retrieval) and retrieval-augmented (RAG) configurations using LLM-as-Judge scoring with Kimi-K2-Thinking.

Baseline Performance: Without access to ETH-specific documentation, Apertus-8B achieves an aggregate score of 0.270 (48 correct or partially correct answers), while Qwen3-8B reaches 0.295 (50 correct or partially correct answers). Baseline responses are typically generic or abstain from answering, reflecting the absence of institution-specific knowledge rather than model failure.

RAG-Enhanced Performance: Providing retrieved archival context substantially improves performance, with Apertus-8B reaching 0.460 (+0.190; 81 correct or partially correct answers) and Qwen3-8B reaching 0.438 (+0.143; 76 correct or partially correct answers).

Figure 2 presents a head-to-head comparison of baseline and RAG-enhanced configurations. Both models benefit significantly from retrieval, confirming that access to institutional documentation is a primary driver of improved answer quality. Differences in RAG effectiveness appear to stem from how retrieved context is incorporated during generation rather than from retrieval quality alone.

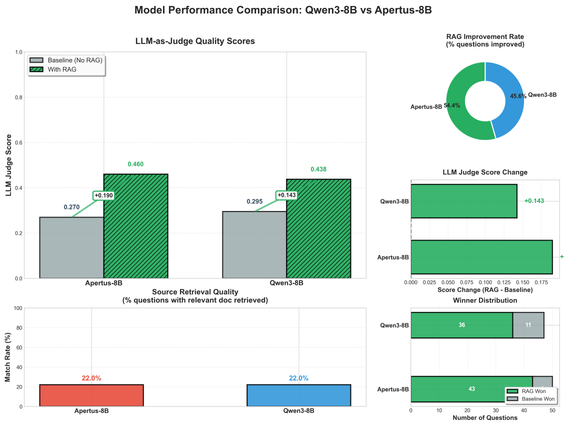

Figure 2: Model Performance Comparison: Qwen3-8B vs Apertus-8B. Shows LLM-as-Judge quality scores, source retrieval quality, RAG improvement rates, and winner distribution between baseline and RAG configurations.

5.2 Multilingual Performance

Figure 3 reveals pronounced language-dependent effects. German-language questions show a higher RAG improvement rate (94.1%) than English questions (71.2%). Retrieval success differs even more strongly: 47.1% of German queries retrieve relevant documents, compared to only 9.1% for English queries.

This asymmetry likely reflects the composition of the ETH web archive, which predominantly contains German-language administrative content. The results highlight the importance of language-aware indexing and retrieval strategies for multilingual institutional archives.

5.3 RAG Improvement Distribution

A more detailed per-question analysis of RAG improvements is provided in Appendix 5. For Apertus-8B, RAG improves answers in 43 cases while degrading performance in only 7, with 50 ties. Qwen3-8B shows 36 improvements, 11 degradations, and 53 ties.

Correct or partially correct answer counts increase substantially with RAG: from 48 to 81 for Apertus-8B (+69%), and from 50 to 76 for Qwen3-8B (+52%). Source retrieval analysis shows both models retrieve reference-aligned documents at similar rates (22%), indicating that differences in RAG effectiveness arise primarily from context utilization during generation rather than from retrieval quality itself.

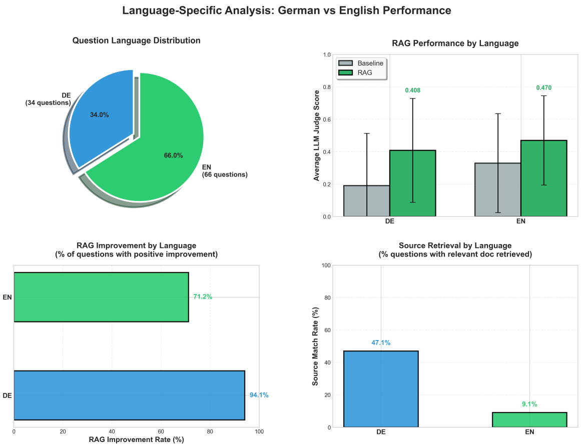

Figure 3: Language-Specific Analysis comparing German (DE) and English (EN) performance. Shows question distribution, RAG performance by language, improvement rates, and source retrieval success rates.

6 Error Analysis

Beyond quantitative metrics, we analyze systematic failure modes through manual inspection of retrieved contexts and generated responses. These errors help explain both the moderate retrieval success rates observed during evaluation and the strong language-dependent performance differences.

A primary issue is the dominance of boilerplate-heavy content in retrieved results. Navigation menus, footer text, and repeated administrative templates often appear as highly similar chunks across many pages. Although sentence-aware chunking improves retrieval granularity, it can amplify this effect by isolating structurally similar segments that are semantically uninformative. As a result, retrieved contexts may be topically related but provide little factual support for answering specific queries.

A second recurring failure mode involves temporal misalignment. Queries referring to specific organizational states or historical policies occasionally retrieve semantically relevant documents from incorrect time periods. While time-aware deduplication reduces redundancy, temporal constraints are not currently enforced during retrieval, leading to outdated or prematurely retrieved content in some cases.

Third, dense similarity scores produced by the embedding model often exhibit limited separation among top-ranked results. We observe that absolute similarity values are relatively close even when contextual relevance differs substantially, particularly in noisy archival corpora. This limits the effectiveness of threshold-based filtering and increases reliance on relative ranking alone.

Finally, retrieval quality degrades for questions requiring aggregation across multiple documents. This suggests that retrieval-augmented generation in archival settings remains sensitive to corpus fragmentation and lacks mechanisms for structured evidence aggregation.

7 Discussion and Future Work

Results demonstrate that retrieval augmentation substantially improves institutional question answering, but absolute performance remains moderate, indicating headroom for improvement. Differences in RAG effectiveness between models appear to stem from how retrieved context is incorporated during generation rather than from retrieval quality alone.

Future work includes improved chunking strategies that reduce boilerplate dominance, metadata-aware retrieval mechanisms that enforce temporal constraints, and lightweight reranking to better separate semantically similar candidates. In addition, evaluation on larger and more systematically curated query sets will be pursued as indexing and retrieval stability improves, enabling more fine-grained analysis of failure modes.

8 Conclusion

We document a data-centric process for transforming a large institutional web archive into a corpus suitable for retrieval-augmented language models. A substantial portion of project effort was devoted to achieving stable, reproducible execution across heterogeneous environments, constraining extensive evaluation. By reporting both system challenges and empirical behavior, this work provides practical guidance for applying RAG to archival web data.

A Ethical Risks

B Ethical Risks

We identified ethical risks related to the use of retrieval-augmented language models over large-scale institutional web archives, particularly concerning the use of generated outputs as potentially authoritative or up-to-date information.

Risk description. The primary risk is that users may rely on generated answers for administrative or procedural decisions without fully accounting for the archival and temporally heterogeneous nature of the underlying data. The main stakeholders affected are institutional users such as students, staff, or administrators, who may receive outdated or incomplete information. A secondary stakeholder is the institution itself, which may face reputational or operational consequences if incorrect guidance is propagated. The severity of individual errors is moderate, but the likelihood of occurrence is non-trivial given observed retrieval of temporally misaligned or boilerplate-heavy content.

Risk evaluation. This risk was evaluated through qualitative inspection of retrieved documents and generated responses during system evaluation. Error analysis revealed recurring failure modes, including retrieval of outdated documents and semantically related but contextually irrelevant content. We also observed strong language-dependent differences in retrieval success, which increases the risk of incomplete answers for certain user groups.

Risk mitigation. We addressed this risk primarily through system design and reporting choices. Retrieved documents retain original timestamps and source metadata, enabling traceability of generated answers. The system is explicitly positioned as a research and analytical tool rather than an authoritative decision-making system. However, fully mitigating temporal misalignment and coverage gaps was beyond the scope of this project, as it would require additional metadata curation and user-facing uncertainty indicators. These limitations are identified as priorities for future work.

C Response Type Analysis

To further characterize baseline behavior, we analyze response types for ETH-specific questions without retrieval. Figure 4 shows the distribution of correct answers, generic responses, refusals, and hallucinations across multiple models.

Larger proprietary models (e.g., Claude Sonnet 4.5 and GPT-5.2) predominantly refuse to answer, explicitly acknowledging missing knowledge. In contrast, APERTUS-8B and QWEN3-8B more frequently produce generic responses. Across all evaluated models, hallucinated answers are rare (0–0%), indicating conservative behavior in the absence of grounding. While generic responses are safer than hallucinations, they rarely provide useful institutional information, motivating retrieval augmentation.

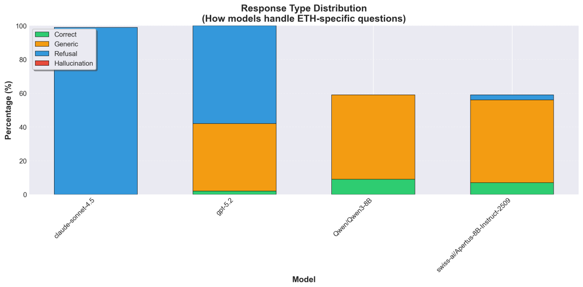

Figure 4: Response Type Distribution across models without RAG. Shows how different models handle ETH-specific questions: Correct (green), Generic (orange), Refusal (blue), Hallucination (red).

References

Patrick Lewis, Ethan Perez, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*.

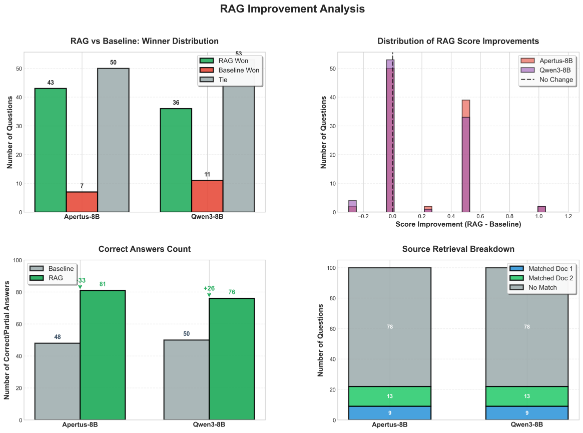

Figure 5: RAG Improvement Analysis showing winner distribution, score improvement distribution, correct answer counts, and source retrieval breakdown for both models.