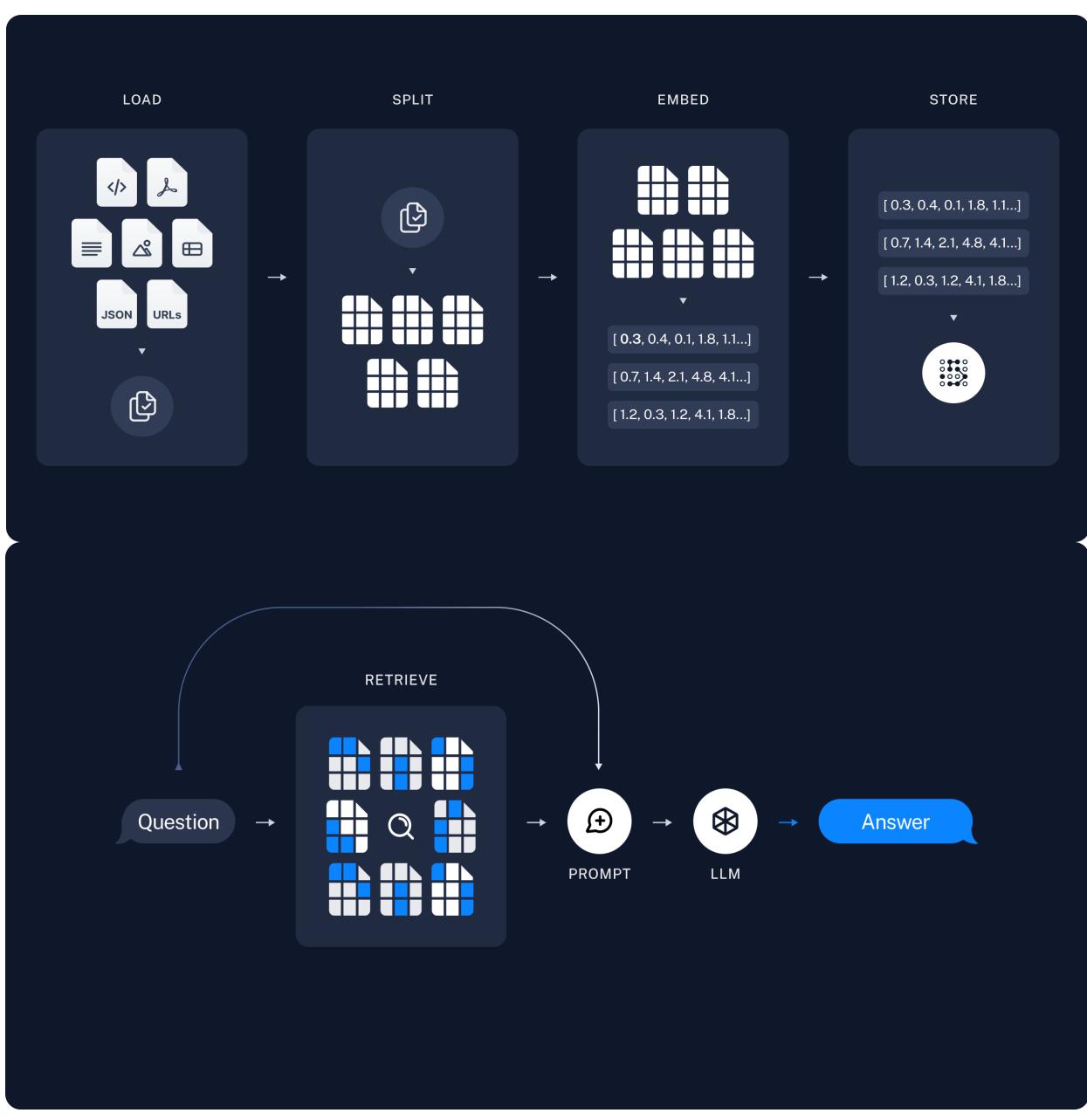


ABSTRACT

The University of Utah's Center for High Performance **Computing (CHPC)** maintains extensive documentation, making it challenging for users to find specific information efficiently. We developed a Retrieval-Augmented Generation (RAG) system to provide an AI-powered question-answering assistant that utilizes local components, leveraging CHPC infrastructure and hardware. This assistant accurately answers user queries via a web API, retrieving relevant documentation chunks and generating context-aware responses, improving knowledge accessibility for CHPC users.

INTRODUCTION

Problem: Users (researchers, faculty, students) need quick, accurate answers to specific questions about HPC resources, software, and procedures. Searching documentation manually can be time-consuming and inefficient. Support staff resources are valuable.



RAG Schematic and Pipeline, LangChain Retrieved May 6, 2025, from https://langchain.com

Leveraging Retrieval-Augmented Generation with Local LLMs for HPC **Documentation Q&A** Kai Ebira, Martin Cuma **Center for High Performance Computing** University of Utah METHODS CONCLUSIONS We successfully developed a Retrieval-Augmented Generation **Data Collection** chatbot to answer user queries about CHPC documentation. Web scraping CHPC documentation using Python (BS4) The system provides rapid, contextually relevant answers grounded in official documentation, reducing user search time **Data Processing** and potentially easing the load on support staff. Files are ingested into a vector database (Qdrant) using This approach demonstrates the potential for localized AI semantic embeddings. assistants to improve access to complex technical information URLs were systematically preserved as metadata associated within HPC environments. with each data chunk during the ingestion process. **Query Processing CHALLENGES/FUTURE WORK** Challenges API Server (Starlette/Uvicorn) RAG Pipeline (LangChain) Model Selection and Optimization: • Context • Retrieval • Inference (Ollama) candidates across diverse architectures and **User Interface** inference. GPT-like chat interface (Django/Python) Supports multiple simultaneous users response quality was suboptimal, exhibiting hallucinations even after tuning inference parameters. RESULTS **CHPC** Assistant **Reset Chat** outputs and hallucinations. Hardware Compatibility and Performance: How can I share data from CHPC to collaborators outside the university? • A key challenge involved choosing appropriate o share data with collaborators outside the university, you can use tools like WinSCP (on /indows) or CyberDuck (on Macs) to drag and drop files. Alternatively, you can use Linux hardware to match the computational needs of the ommands such as <code>scp</code> , <code>rsync</code> over <code>ssh</code> , <code>sftp</code> , <code>wget</code> , or <code>curl</code> on CHPC interactive nodes or Data LLM used. For getting a download link for data on a website like Dropbox or Sharefile, use the Chrome extension CurlWget or the Firefox extension cliget to generate a URL usable with wget or curl. You can also use FastX or Microsoft Remote Desktop to access a host and then use a web browser (or Box Sync) to transfer files to/from Box.com, which is the only acceptable UofU cloud storage GPU yielded baseline performance metrics. Finally, CHPC PIs can set up guest accounts for collaborative file sharing without creating a full CHPC account; see https://guest-transfer.chpc.utah.edu/. resulted in a significant reduction in inference https://www.chpc.utah.edu/documentation/data_services.php .chpc.utah.edu/documentation/policies/1.6SecurityPolicy.php#Pol1.6.4 latency, decreasing by over 50%. https://www.chpc.utah.edu/resources/hosting.php • Future Work: Integration with Open OnDemand • Systematic evaluation framework The chatbot successfully cites passages from CHPC documentation and even includes sources for user convenience.



 Identifying an optimal open-source Large Language Model (LLM) necessitated evaluating numerous parameter sizes to achieve efficient and accurate

• The initial prototype utilized Llama 3.2. However,

 Transitioning to Gemma 3 demonstrably improved performance, significantly reducing undesired

 Initial experiments conducted on an AMD MI100 Subsequent migration to an Nvidia RTX A4000 GPU

• UI enhancements (e.g., feedback mechanism).



