Leveraging Retrieval-Augmented Generation with Local LLMs for HPC Documentation Q&A

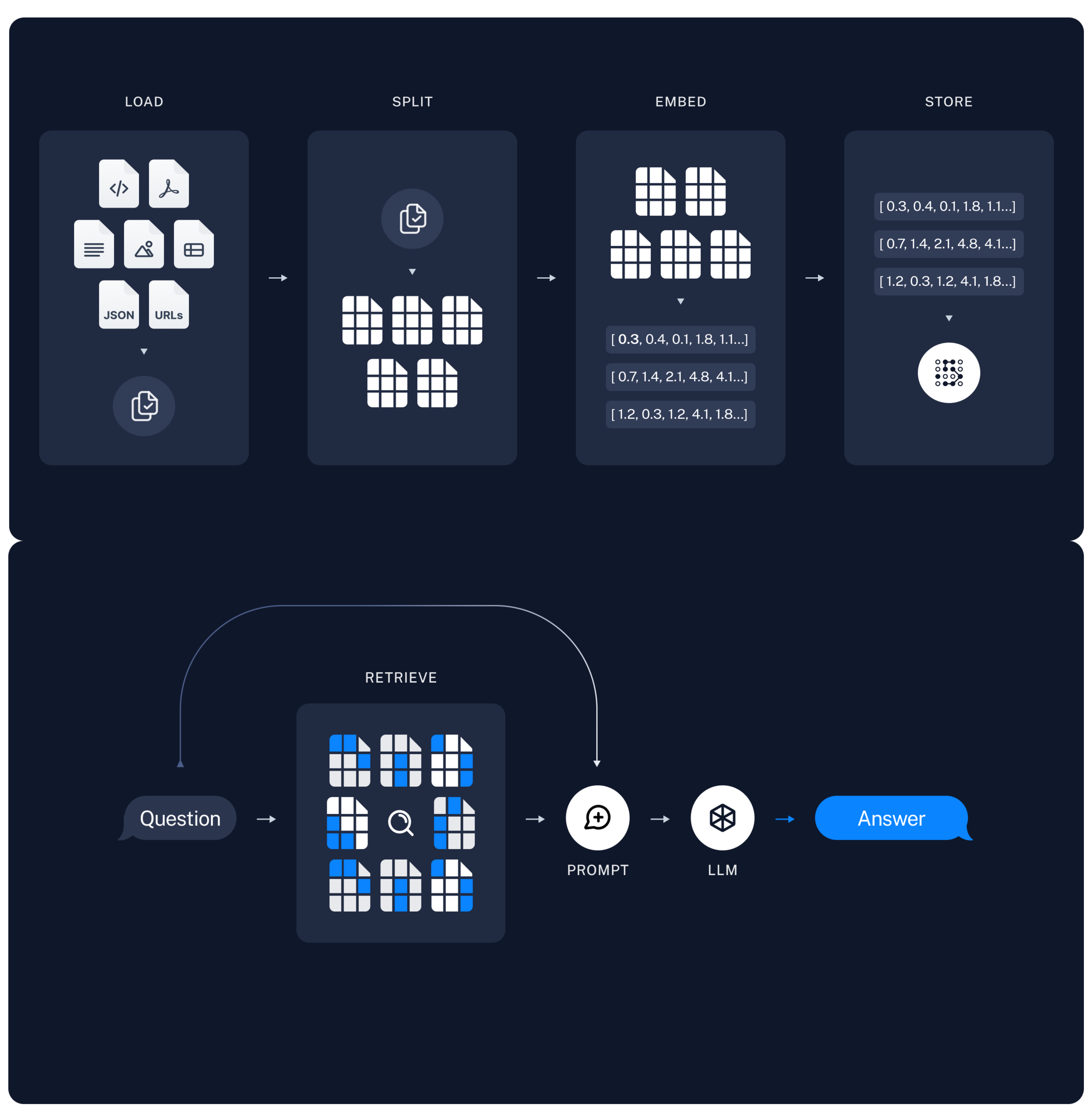
Kai Ebira, Martin Cuma
Center for High Performance Computing
University of Utah

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`

METHODS

Data Collection

- Web scraping CHPC documentation using Python (BS4)

Data Processing

- Files are ingested into a vector database (Qdrant) using semantic embeddings.
- URLs were systematically preserved as metadata associated with each data chunk during the ingestion process.

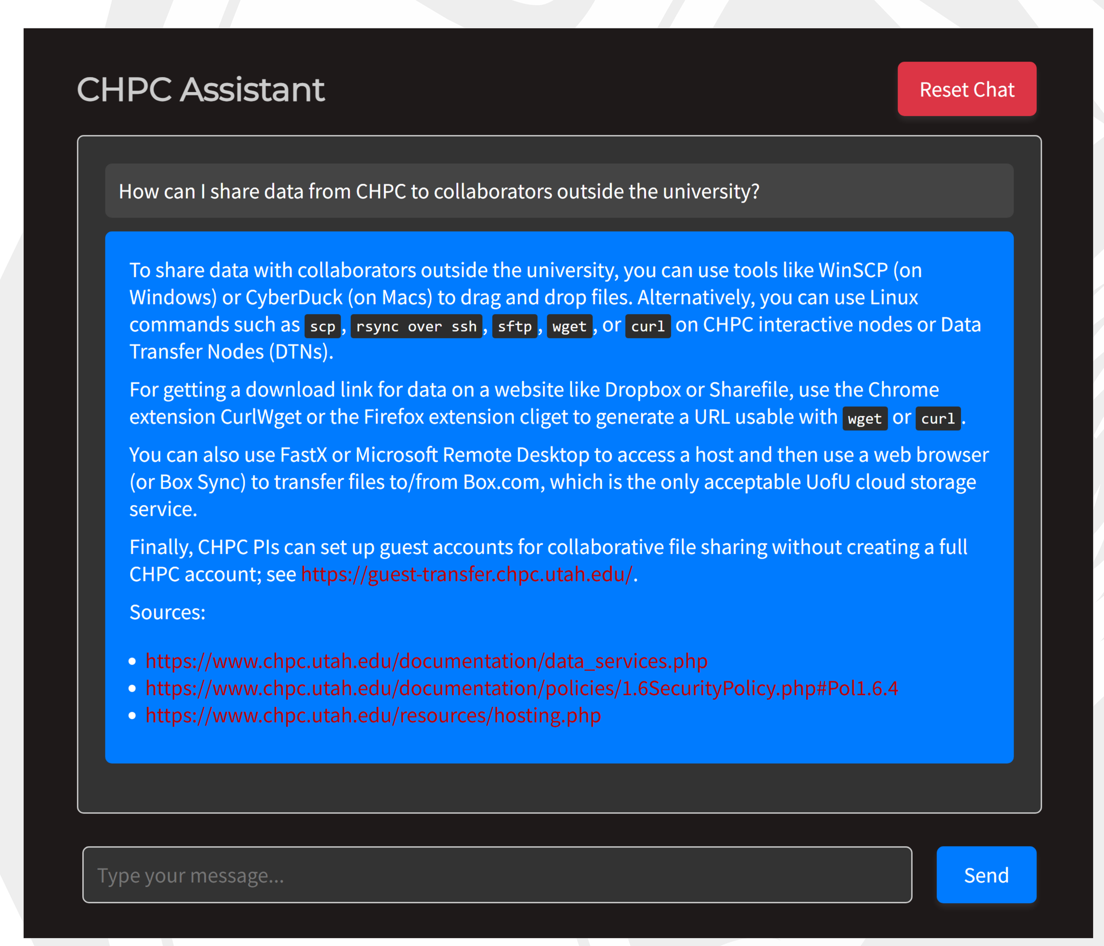
Query Processing

- API Server (Starlette/Uvicorn)
- RAG Pipeline (LangChain)
  - Context
  - Retrieval
  - Inference (Ollama)

User Interface

- GPT-like chat interface (Django/Python)
- Supports multiple simultaneous users

RESULTS



The chatbot successfully cites passages from CHPC documentation and even includes sources for user convenience.

CONCLUSIONS

We successfully developed a Retrieval-Augmented Generation chatbot to answer user queries about CHPC documentation.

The system provides rapid, contextually relevant answers grounded in official documentation, reducing user search time and potentially easing the load on support staff.

This approach demonstrates the potential for localized AI assistants to improve access to complex technical information within HPC environments.

CHALLENGES/FUTURE WORK

Challenges

- **Model Selection and Optimization:**
  - Identifying an optimal open-source Large Language Model (LLM) necessitated evaluating numerous candidates across diverse architectures and parameter sizes to achieve efficient and accurate inference.
  - The initial prototype utilized Llama 3.2. However, response quality was suboptimal, exhibiting hallucinations even after tuning inference parameters .
  - Transitioning to Gemma 3 demonstrably improved performance, significantly reducing undesired outputs and hallucinations.
- **Hardware Compatibility and Performance:**
  - A key challenge involved choosing appropriate hardware to match the computational needs of the LLM used.
  - Initial experiments conducted on an AMD MI100 GPU yielded baseline performance metrics. Subsequent migration to an Nvidia RTX A4000 GPU resulted in a significant reduction in inference latency, decreasing by over 50%.

- **Future Work:**
  - UI enhancements (e.g., feedback mechanism).
  - Integration with Open OnDemand
  - Systematic evaluation framework