

Technical Report

Sanskrit Document Retrieval-Augmented Generation (RAG) System

1. Introduction

This project presents the design and implementation of a Retrieval-Augmented Generation (RAG) system for answering queries based on Sanskrit documents, developed as part of a company-assigned technical assessment. The objective of the system is to retrieve relevant contextual information from Sanskrit textual data and generate coherent, context-grounded responses using a CPU-only inference pipeline. The entire system is executed locally using VS Code and adheres strictly to the CPU-only constraint specified in the assignment.

2. System Architecture and Flow

The system follows a standard Retrieval-Augmented Generation architecture, where document retrieval and text generation are clearly separated into modular components. The pipeline consists of document loading, preprocessing and chunking, vector embedding, vector-based retrieval, context-aware generation, and an interactive query interface. This modular separation ensures clarity, scalability, and ease of maintenance.

3. Sanskrit Documents Used

The system uses only the Sanskrit documents provided by the company as part of the assignment. These documents consist of classical Sanskrit prose and narrative texts, including moral stories and illustrative examples. No external datasets or annotated corpora were used.

4. Preprocessing Pipeline for Sanskrit Documents

The preprocessing pipeline includes Unicode-safe text loading, whitespace normalization, and chunking with overlap. Chunking is essential to preserve semantic coherence, prevent information loss, and enable efficient vector retrieval. Sentence-level tokenization was avoided due to inconsistencies in Sanskrit punctuation.

5. Retrieval and Generation Mechanisms

A vector-based retrieval approach was implemented using multilingual sentence embeddings and a CPU-based FAISS index. Relevant document chunks are retrieved using similarity search. The generation component uses a lightweight instruction-tuned language model optimized for CPU inference. The model generates responses strictly grounded in the retrieved Sanskrit context.

6. Performance Observations

All experiments were conducted on a local CPU environment. Document loading and preprocessing complete within seconds. Initial embedding computation takes a few minutes, while query retrieval occurs in under a second. Response generation typically completes within 10–20 seconds. The system demonstrates reliable relevance, controlled hallucination behavior, and efficient resource usage under CPU-only constraints.

7. Conclusion

This project demonstrates a complete, modular, and efficient Sanskrit Retrieval-Augmented Generation system operating entirely on CPU. By combining vector-based retrieval with controlled text generation, the system provides accurate and context-aware answers grounded in Sanskrit documents and satisfies all assignment requirements.