📜 Sanskrit RAG System - Project Report

Project: Retrieval-Augmented Generation for Sanskrit Documents

Name: Rudraksh seth

Overview

The Sanskrit RAG System enables intelligent semantic search and answer generation from Sanskrit documents using CPU-optimized models. Users query in Sanskrit or English and receive contextually grounded answers in pure Sanskrit, supporting heritage language preservation without GPU infrastructure.

Technology Stack

Component	Technology
UI	Streamlit
Embedding	Sentence-Transformers MiniLM (50+ languages)
Search	FAISS IndexFlatL2 (CPU-optimized)
Generation	Sarvam-1 LLM (Sanskrit-first)
Prompt	LangChain ChatPromptTemplate

Key Achievements

1. ChatPromptTemplate with Sanskrit Rules
7-rule enforcement system in Sanskrit ensuring context-grounded answers. Strict language policy: Pure Sanskrit output only (no English). Single-sentence responses with proper Devanagari compliance.

2. CPU-Optimized Architecture
384-dim multilingual embeddings (~420MB). FAISS exact L2 matching on CPU. ~2-3GB peak memory, no GPU required. 1-2 second query response time.

3. Sanskrit-Aware Processing
RecursiveCharacterTextSplitter with linguistic separators (॥ ।). 500-char chunks with 150-char overlap. Preserves semantic boundaries.

4. Multi-Script Query Support
Devanagari detection and English flexibility. Query type identification for routing. Minimum 3-character validation.

5. Persistent Vector Indexing
FAISS indexes saved to disk. Real-time document addition. Multiple concurrent stores.

Results

✅ Fully functional Streamlit application

✅ Multi-format support (PDF, DOCX, TXT)

✅ Pure Sanskrit generation in Devanagari

✅ CPU-efficient (no GPU needed)

Configuration

Chunk Size: 500 characters | Overlap: 150 characters
Retrieved: Top 4 documents | Max Tokens: 50
Languages Supported: 50+ (including Sanskrit)
Load Time: ~1-2 min (first run) | Query Speed: 30 sec

Use Cases

Educational: Classical Sanskrit text learning

Research: Manuscript analysis and accessibility

Preservation: Digital archival of ancient literature

Cultural Heritage: Language revival initiatives

Conclusion

Demonstrates sustainable Sanskrit language processing combining ChatPromptTemplate rule enforcement with CPU-specific model selection. Provides practical framework for accessing Sanskrit textual heritage without expensive GPU infrastructure.

Stack: Python, Streamlit, LangChain, FAISS, Transformers