Sanskrit RAG System – Aseem

A Retrieval-Augmented Generation (RAG) pipeline built for Sanskrit texts, running entirely on CPU.

1. Features

- Load and preprocess Sanskrit documents
- Chunking with overlap
- Two retrievers:
  - TF-IDF (baseline)
  - SBERT embedding-based retriever (final)
- Lightweight generator for answer synthesis
- Full CLI support (`app.py`)

2. Installation

Create Conda Environment

```
conda create -n rag_env python=3.11 -y
conda activate rag_env
```

Install Dependencies

```
conda install pytorch torchvision torchaudio -c pytorch -c
conda-forge -y
pip install sentence-transformers scikit-learn numpy
```

3. Preprocess the Data

```
python3 code/preprocess.py
```

4. Build Embeddings (recommended retriever)

```
python3 code/build_embeddings.py
```

5. Run RAG System

Embedding Retriever:

```
python3 code/app.py --query "मूर्खभृत्यस्य उपदेशः कः ?" --use-emb
```

TF-IDF Retriever:

```
python3 code/app.py --query "मूर्खभृत्यस्य उपदेशः कः ?"
```

6. Files & Directories

```
code/                → all Python scripts
data/                → Sanskrit .txt, chunks, embeddings
report/              → final_report.pdf
README.md            → this file
requirements.txt
```

7. Notes

- Entire pipeline runs on CPU
- Supports any Sanskrit `.txt` document
- Embedding retriever gives best accuracy