Retrieval-Augmented Generation (RAG) Solution Documentation

Overview of Solution Components

Component	Function
Embedding Model	Converts text to embeddings for similarity search using `Alibaba-NLP/gte-large-en-v1.5`.
Document Parser	Parses PDF and DOCX documents using Langchain's `PyPDFLoader` and `Docx2txtLoader`.
Content Chunker	Splits documents into chunks using `RecursiveCharacterTextSplitter` from Langchain.
Vector Database	Stores and retrieves document embeddings using Chroma DB.
Retrieval Algorithm	Performs similarity search and re-ranks results using `BAAI/bge-reranker-v2-m3`.

Component	Function
Output Formatter	Formats and outputs the results in CSV format.

Workflow Review

High-Level Workflow

1. **Query Input:** User submits a query.
2. **Document Parsing:** Input documents (PDF and DOCX) are parsed into text using Langchain's `PyPDFLoader` and `Docx2txtLoader`.
3. **Content Chunking:** Parsed text is split into chunks using `RecursiveCharacterTextSplitter` with a chunk size of 2000 characters and overlap of 300. Custom separators are used to maintain context, and short chunks are merged during post-processing.
4. **Embedding Generation:** Chunks are converted to embeddings using `Alibaba-NLP/gte-large-en-v1.5`.
5. **Indexing:** Embeddings are indexed in Chroma DB.
6. **Retrieval:** Top 20 similar chunks are retrieved using Chroma DB's similarity search.
7. **Re-ranking:** Retrieved chunks are re-ranked using `BAAI/bge-reranker-v2-m3`.
8. **Output Formatting:** Top 5 relevant chunks are formatted into CSV.

Key Advantages

- **Utilization of Robust Models:** Uses state-of-the-art embedding and re-ranking models from HuggingFace with a reasonable size.
- **Effective Document Parsing:** Efficiently handles both PDF and DOCX formats using Langchain.
- **Advanced Chunking Strategy:** Employs recursive character text splitting with custom separators and post-processing to maintain context and avoid very small chunks.
- **Open-Source Tools:** Entirely built with open-source tools ensuring transparency and reproducibility.

- **Local and Offline Capability:** Designed to run fully locally.

Requirements and Dependencies

- **Python 3.8+**
- **Libraries:**
  - `langchain`
  - `chromadb`
  - `huggingface-hub`
  - `sentence-transformers`
  - `pandas`
  - `numpy`
  - `transformers`
  - `xformers`

Installation and Usage Guidance

Installation Steps

1. **Create a Virtual Environment:**

```
python3 -m venv venv
source venv/bin/activate
```

2. **Create a Virtual Environment:**

```
pip install -r requirements.txt
```

3. **Download and Set Up Models:** From HuggingFace, download and set up the Alibaba-NLP/gte-large-en-v1.5 and BAAI/bge-reranker-v2-m3 models locally.

Usage Instructions

1. **Prepare Your Documents:** Place all your PDF and DOCX documents in the documents directory.

Troubleshooting Guide - Common Issues and Solutions

Dependencies not installed correctly

Solution: Ensure you are in the virtual environment and run `pip install LIBRARY_NAME` again.

Model not found

Solution: Verify the model download path.

Incorrect document parsing

Solution: Check document formats and ensure they are placed correctly in the documents directory.

Additional Information

Performance Tips

- Use a GPU for faster embedding generation and retrieval if available.
- Ensure documents are well-formatted and clean for better parsing and chunking.

Future Enhancements

- Implement query pre-processing techniques to improve input query quality.
- Fine-tune the embedding model with domain-specific data.
- Explore adaptive chunking strategies based on document structure.
- Introduce multi-stage re-ranking mechanisms.