

**Step 1: Document Processing and Encoding**

This phase focuses on preparing documents for retrieval.

1. **PDF Document**: The system starts with a document file (PDF, etc.).
2. **Document Loader**: This component loads the document into the system.
3. **Document Text**: The text content of the document is extracted.
4. **Text Splitter**: The document text is divided into smaller pieces (chunks), making it easier to search later.
5. **Document Chunks**: The result of splitting the text into manageable chunks.
6. **Chunk Embeddings**: Each chunk of text is transformed into a vector representation (embedding) using an encoder model. These embeddings capture the semantic meaning of the text.
7. **Vector DB (Database)**: These chunk embeddings are stored in a vector database, where similar embeddings can be efficiently searched and retrieved.

**Step 2: Query-Answer Search and Generation**

This phase handles user queries and generates answers using document chunks.

1. **User Query**: A user inputs a query into the system.
2. **LLM Embedder**: The user query is processed by a **Large Language Model (LLM) Embedder**, which converts the query into an embedding (similar to the chunk embeddings).
3. **Query Embedding**: The query is transformed into its vector representation (embedding) for search purposes.
4. **DB Retriever**: The query embedding is sent to the vector database (from Step 1), which retrieves the most similar document chunks.
5. **Similar Document Chunks**: The most relevant document chunks related to the query are retrieved.
6. **Prompt Template**: These retrieved document chunks are formatted into a prompt that will be sent to the LLM.
7. **Prompt Template**: This step involves assembling the query with the retrieved document chunks to create a coherent prompt.
8. **LLM Prompt**: The formatted prompt, consisting of the query and related document chunks, is passed to the LLM for response generation.
9. **LLM**: A large language model (e.g., GPT) generates a response to the query based on the retrieved document chunks and the query itself.
10. **Generated Response**: The LLM produces a human-like answer to the query.
11. **Generated Response**: The final response is presented to the user.

Code skeleton

Install --

pip install transformers sentence-transformers faiss-cpu

Step 1: Document Processing and Embedding