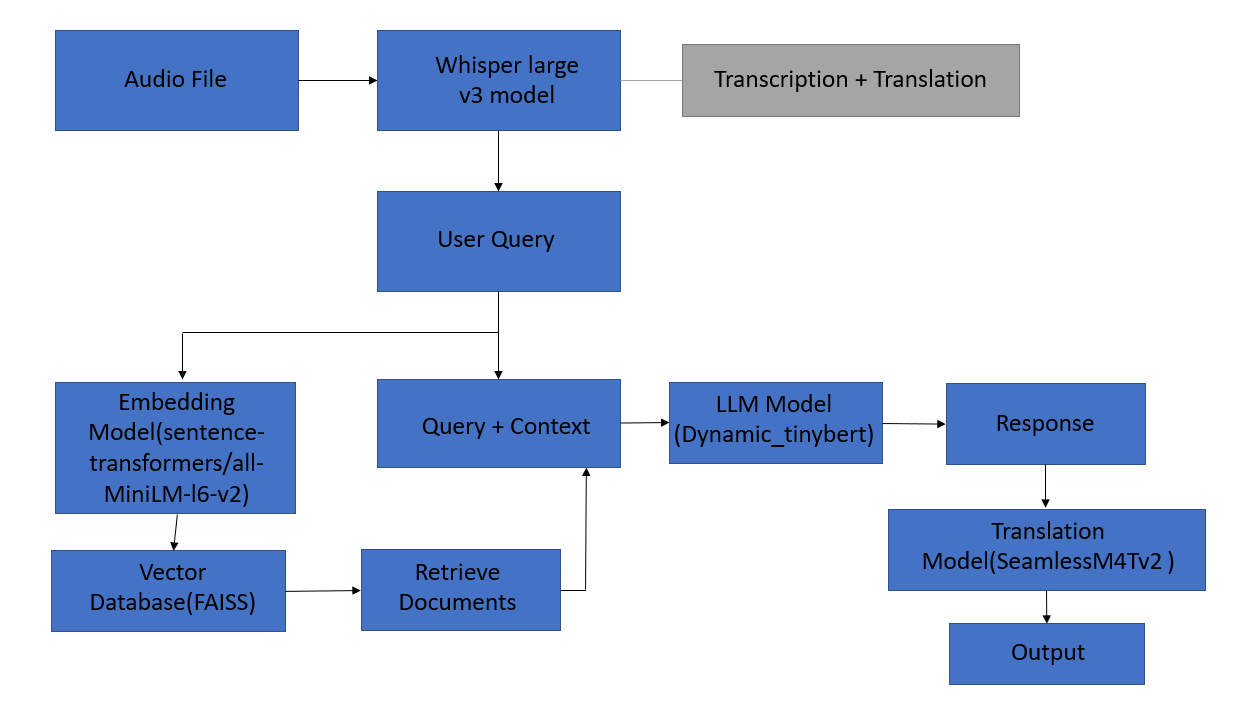
**Building a Multilingual Speech Recognition Model for RAG Without Training**

The use of LangChain and Hugging Face libraries to build a system that combines pre-trained dense retrieval and sequence-to-sequence models for generating responses.

**Workflow:**

****

**Document Loading:**

Utilizes the Hugging Face dataset "databricks-dolly-15k" for instruction-following records.

Implements the HuggingFaceDatasetLoader to load data from the specified dataset and column.

**Text Splitting:**

Uses the RecursiveCharacterTextSplitter from LangChain to split long documents into smaller, manageable chunks.

**Text Embedding:**

Employs the HuggingFaceEmbeddings class to capture semantic meaning and generate embeddings for the text.

**Vector Stores:**

Utilizes the FAISS vector store for efficient storage and searching of the generated embeddings.

**Searching Relevant Documents:**

Performs a similarity search for a given question using the FAISS vector store.

**Preparing the LLM Model:**

Chooses the Intel/dynamic\_tinybert model for question-answering.

Creates a question-answering pipeline using Hugging Face, extending its functionality with a LangChain pipeline.

**Retrievers:**

Sets up a retriever interface to retrieve relevant documents from the database based on queries.

**Retrieval QA Chain:**

Creates a RetrievalQA chain that combines question-answering with a retrieval step, using the language model and vector database.

**Conclusion:**

Highlights the fun and informative experience of implementing RAG with open-source libraries like LangChain and Hugging Face.

Mentions LangChain as an open-source developer framework for building large language model applications.