LearnTube - YouTube QA Bot for Lecture Videos

Prepared by: Reema Alabisi - Sahar Sheshah

##Project Goal

This project empowers students to interactively explore educational YouTube videos using the **LearnTube** platform. It transforms lectures into dynamic learning experiences by:

- Automatically analyzing video content
- Generating questions and summaries
- Offering real-time interactions with the content
- Supporting multilingual translations for greater accessibility

Starting with videos on Artificial Intelligence, the platform is designed to enhance comprehension and engagement, helping learners absorb complex topics more effectively-all powered by advanced AI technologies.

##Description

LearnTube is an AI-powered chatbot designed to enhance the way students engage with educational content on YouTube. By extracting audio from videos and transcribing it using OpenAI's Whisper model, the system builds a structured, searchable knowledge base that supports interactive learning.

Users can input natural language questions related to the video content and receive accurate, context-aware answers—thanks to the integration of **LangChain** and OpenAI APIs. This ensures a smooth and intelligent Q&A experience that stays grounded in the video material.

To showcase its functionality, **LearnTube** currently focuses on a curated selection of YouTube videos about **Artificial Intelligence**, allowing learners to explore complex AI concepts through real-time dialogue, summaries, and translations. The platform is also adaptable to a wide range of subjects, making it a versatile tool for students and lifelong learners.

AI Techniques Used:

- **Speech-to-Text Conversion:** OpenAI Whisper.
- **Natural Language Processing:** OpenAI models + LangChain for QA and summarization.
- **Multimodal Interface:** text input.
- **Vector Search:** Pinecone to find relevant content.
- **LangChain Agents:** Managing tool routing and response generation.

##🎓 Potential Applications:

Educational Companion: Helps students learn from lectures and tutorials.

Topic-Focused Review: Especially useful for complex subjects like AI.

Flexible Learning Aid: Supports text-based learning styles.

##Key Features

📽️ **Transform Educational Videos** into interactive, AI-powered learning sessions

🤖 **Integrate Artificial Intelligence** to personalize and enhance the learning experience

🧠 **Simplify Complex Topics** using summaries, Q&A, and real-time interaction

🌐 **Enable Multilingual Translation** to make content accessible to a global audience

⏱️ **Support Self-Paced Learning** by allowing students to explore content at their own speed

Architecture Overview

playlist_links = ['https://www.youtube.com/playlist?list=PLdKd-j64gDcDVXmhHLIRIqpfnxiJadMjd']

audio_inputs_path: Directory where extracted audio files are saved.

save_transcript_path: Directory where the transcribed text files are stored

Core Components

1. 📽️ Video Processing Pipeline

- YouTube video downloader (yt-dlp)
- Audio extraction and chunking (pydub)
- Whisper-based transcription (transformers)

2. 📝 Text Processing

- Transcript chunking (RecursiveCharacterTextSplitter)
- OpenAI embeddings for vectorization

3. 🔍 Retrieval-Augmented Generation

The system relies on **LangChain RetrievalQA** to answer questions based strictly on the video transcript, ensuring that all responses are accurate and grounded in the original content.

4. 💬 Conversational Interface

A conversational interface built using **LangChain Agent** with memory support to enable smooth and natural multi-turn interactions.
The interface provides two key tools:

- **Question Answering (Q&A):** For directly answering user queries
- **Summarization:** To help users quickly understand the main ideas of the video content
  It also includes **multilingual translation** to improve accessibility for diverse learners.

5. 📊 **Evaluation System**

- **Automated generation of question-answer pairs**
- **Answer accuracy evaluation using LLMs**
- **Faithfulness checks** to ensure responses remain true to the source material
- Faithfulness verification pipeline

💡 **Why This Architecture?**

Every component was chosen for:

- **Reliability:** Each step (downloading, transcription, chunking, embedding, retrieval) is robust and modular.
- **Transparency:** Strict adherence to source content ensures factuality and trust.
- **Scalability:** pinecone and chunked processing enable handling large video libraries.

🧠 **Design Philosophy & System Logic**

1. Transcript-Based Question Answering (Source-Locked Responses)

The system is intentionally designed to generate answers **exclusively from the transcribed content** of the video. By enforcing this strict reliance on the source, **LearnTube** ensures factual consistency and avoids introducing any external or assumed information.

##How it works?
All question-answering chains and conversational prompts are explicitly configured to rely solely on the transcribed content of the video. If the answer is not present in the transcript, the system responds with: "There is no information about that in the videos." instead of making assumptions. This ensures that all responses remain strictly related to the video content.

2. Chunked Processing & Retrieval-Augmented Generation

Long transcripts exceed LLM context windows and reduce retrieval precision. Chunking (using RecursiveCharacterTextSplitter) ensures each vector represents a coherent, retrievable segment.

Transcripts are split into overlapping chunks, embedded, and indexed with Pinecone for fast, accurate retrieval.

3. OpenAI Whisper for Transcription

Whisper provides accurate audio transcription, which is essential for converting educational video audio into searchable text. This is achieved by automatically extracting and transcribing the audio using Whisper,

with a built-in mechanism to handle errors during the process.

###4. Pinecone Vector Database

Pinecone provides a fully managed, cloud-native vector database that enables fast, scalable, and real-time similarity search—ideal for handling large collections of transcript embeddings in QA systems.

###5. Evaluation in RAG System

The evaluation covers:

- ✅ **Answers:** Checked for accuracy and alignment with the transcript
- 📝 **Summaries:** Assessed for clarity and relevance
- 🌍 **Translations:** Reviewed for multilingual accuracy
- 🔒 **Faithfulness:** Ensures all outputs stay true to the video content