Recall : Making Videos Interactive

This article discusses details of our hackathon project for the LLM Agents Berkeley MOOC Fall 2024 course.

The hackathon project's goal is to build a video system capable of supporting chat style interactions with the user. Our narrow focus is on presentation style videos, such as in a conference, workshops or in MOOCs like this LLM Agents Berkeley Course.
In order to accomplish this, we divide the system into two components:

First major agentic component is the video ingestion pipeline. This uses a video model, a vector db, a regular db and a LLM backend in order to convert a given video into a knowledge base.

The second component is the video serving pipeline which works on the precomputed knowledge base from where the context is extracted, user chat history and a backend multimodal LLM.

Video ingestion pipeline:
The video ingestion pipeline is further designed as follows: Since the audio can contain a large amount of context, we use a transcription service based on Whisper to convert the audio into text along with preserving the timestamps. The video is also converted into a series of images to ensure each image is "significantly" different from another. We use OpenCV utilities to accomplish this and we experiment with various possible algorithmic definitions of "significant change" from one image to another. These sets of images, with their time index preserved, are fed into a multimodal LLM to gather a text description. They are also converted into an embedding space using a CLIP model.

This design allows us to extract as much of context as possible across a sequence of images in order to develop a deeper understanding of the video.
Combined with the transcription, these are indexed into a vector db and a regular db for retrieval. Together they comprise the video knowledge base. It is possible to add additional contextual documents to this knowledge base to increase the information available for the system to use. The documents would be indexed into the same vector embedding space as the other assets for efficient retrieval.

Video serving pipeline:

The serving pipeline loads the vector db and related information into an in-memory store from which we retrieve in a RAG manner. We use LlamaIndex for the vector store and the retrieval mechanisms. Given a chat conversation, which could include a question, we use the vector db to find matching context. This design is similar to the one shown in Lecture 4 from Jerry Liu using LlamaIndex for complex document query systems. When the chat and context is presented to the backend LLM, it has the support of tool calling to extract additional images as needed to supplant the generated answer.

We use a secondary LLM phase to determine the best snippets to extract from the video so that a "video answer" can be generated for the query.

The core technical challenges include:

- Lack of a clear eval dataset for the presentation/tutoring video vertical which is significant percentage of all technical/educational videos.
- Lack of a strong multimodal system to maintain context for longer period of time to allow for more complex reasoning.
- Efficient speaker diarization to determine the number of speakers/who is talking.

Our future work includes addresses these challenges to make the system a delight for our users.