File -1: taking a vedio extracting audio file from it and converting into text and summarize

Video to Text Summarization (Static)

Steps

Convert video to audio (Step 1): The video is converted into audio, which is easier to process.

Transcribe audio to text (Step 2): The extracted audio is transcribed into textual content.

Summarize the transcribed text (Step 3): The lengthy transcribed text is summarized, providing a concise representation of the original content.

Required Libraries

moviepy: Used for video processing and audio extraction.

speech\_recognition: Facilitates audio-to-text transcription.

summarizer: Used for text summarization.

Advantages

Multilingual Support: The system can convert audio from videos in various languages into text.

Scalability: It can handle large-sized video files with ease.

User-Friendly: The process is straightforward and user-friendly.

Disadvantages

Language Support: It may not work for languages with limited voice recognition support.

Audio Quality: The accuracy of conversion is highly dependent on the quality of the audio extracted from the video.

File Downloads: You'll need to download specific files (e.g., video, audio) and specify their paths.

Processing Time: The process can be slower for longer videos or lower-quality hardware.

output:

output for write into file statements:

C:\Users\Sivamani\OneDrive\Desktop\Summarizing Audio Files in Python>C:/Users/Sivamani/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Sivamani/OneDrive/Desktop/Summarizing Audio Files in Python/without/vedio\_summarize.py"

MoviePy - Writing audio in without\data\output\_audio.wav

MoviePy - Done.

Downloading (…)lve/main/config.json: 100%|█████████████| 571/571 [00:00<?, ?B/s]

Downloading model.safetensors: 100%|███████| 1.34G/1.34G [05:54<00:00, 3.80MB/s]

Downloading (…)okenizer\_config.json: 100%|███████████| 28.0/28.0 [00:00<?, ?B/s]

Downloading (…)solve/main/vocab.txt: 100%|████| 232k/232k [00:00<00:00, 558kB/s]

Downloading (…)/main/tokenizer.json: 100%|███| 466k/466k [00:00<00:00, 2.00MB/s]

output for print statment

C:\Users\Sivamani\OneDrive\Desktop\Summarizing Audio Files in Python>C:/Users/Sivamani/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Sivamani/OneDrive/Desktop/Summarizing Audio Files in Python/without/vedio\_summarize.py"

MoviePy - Writing audio in without\data\output\_audio.wav

MoviePy - Done.

erase may 4:23 and you are watching the covid Python is a wonderful language for productive program in but has one big problem is too slow and going slow means you get made fun of breast in C plus plus chats of the world but the tables are about to turn thanks to a brand new programming language called mojo a superset of python is not just two times faster not 10 times faster but up to 35 thousand times faster than the dominant language for artificial intelligence but behind the current company founded by Chris programming language to take advantage of language

file :

pip install pydub

git clone https://git.ffmpeg.org/ffmpeg.git ffmpeg