Clip Master: Automated Video Highlights Using AI

Guided By:
Mr Prakash Sinha

Group Members:

Ameya Bhawsar	240340128003
Shrinivas Jawade	240340128010
Kunal Kurve	240340128012
Manasi Malge	240340128013
Pranav Gaddi	240340128019

Project Objective

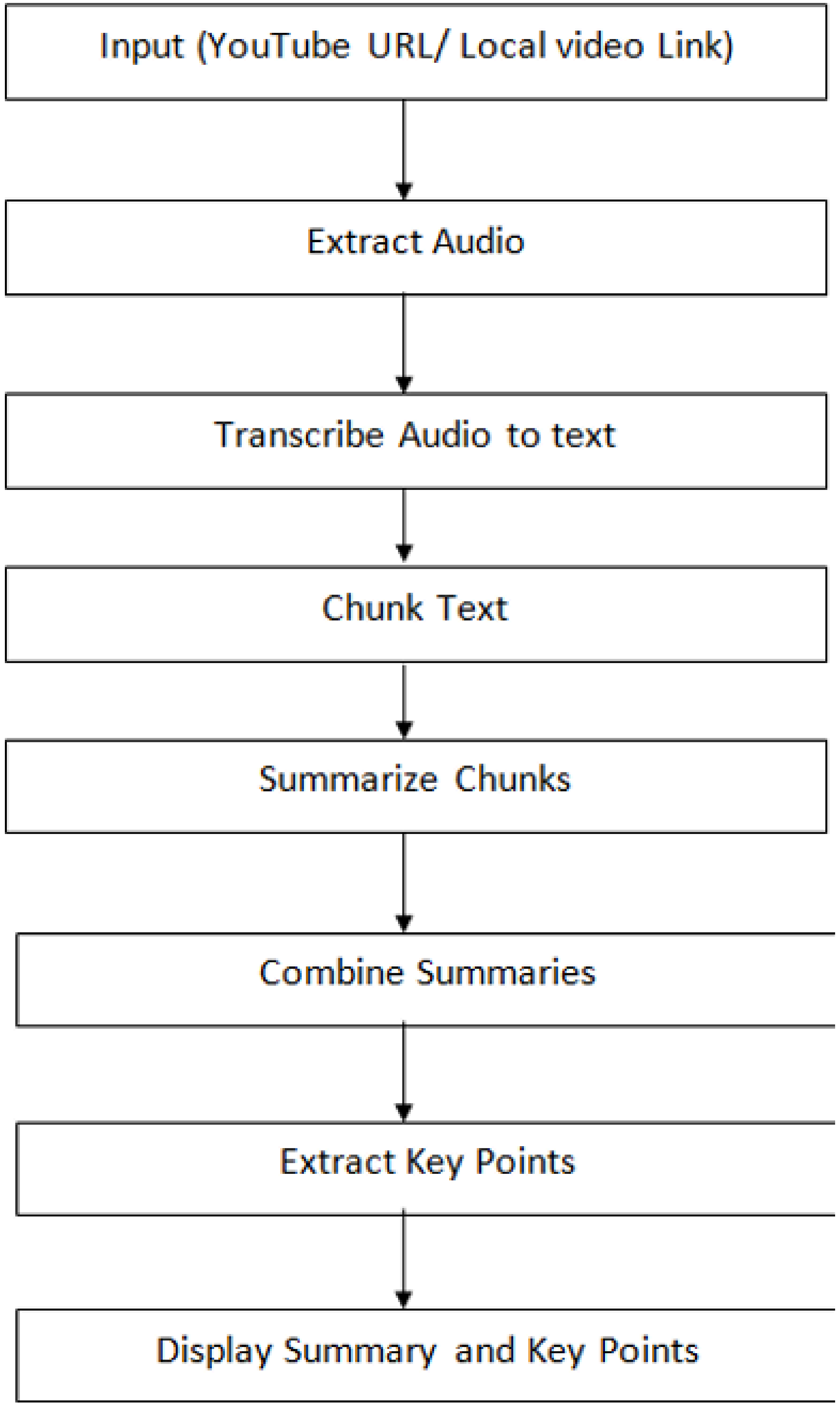
- This project aims to develop an easy-to-use tool that converts videos into concise written summaries.
- Users can upload videos or provide online links, and the tool will automatically generate a summary of the video's key points.
- This helps people quickly grasp important information from long videos, making it especially useful for education, research, and content creation.

Approach

- We create a robust system for transcribing and summarizing video content through advanced deep-learning models and natural language processing techniques.
- The solution is implemented in Python and integrated with a Streamlit web application, enabling users to process both online and local video files.
- Steps to solve the project:
    - Video Acquisition: Upload video files directly or paste video URLs.
    - Audio Conversion: Extract audio tracks from the uploaded or downloaded videos.
    - Speech-to-Text Conversion: Transcription using Whisper.
    - Text Summarization: Using BART Model.
    - Output Presentation: Structure the generated summary in a clear and readable format.

Flow Chart

Given is a flow chart of the model



Input (YouTube URL/ Local video Link)

Extract Audio

Transcribe Audio to text

Chunk Text

Summarize Chunks

Combine Summaries

Extract Key Points

Display Summary and Key Points

Requirements and Specifications

Software Requirements

Python Libraries:

streamlit: For building and running the web-based user interface.
yt-dlp: For downloading audio from YouTube videos.
whisper: For transcribing audio to text.
transformers: Provides access to the BART model for text summarization.
nltk: For tokenizing text into sentences, used in the text chunking process.
Subprocess (built-in): A built-in Python module for running system commands.

Cloud Platforms:

Jupyter Notebook / Jupiter Lab
Google Colab

Hardware Requirements

OS: Windows 11,MAc Os, Ubuntu

RAM: 16 GB

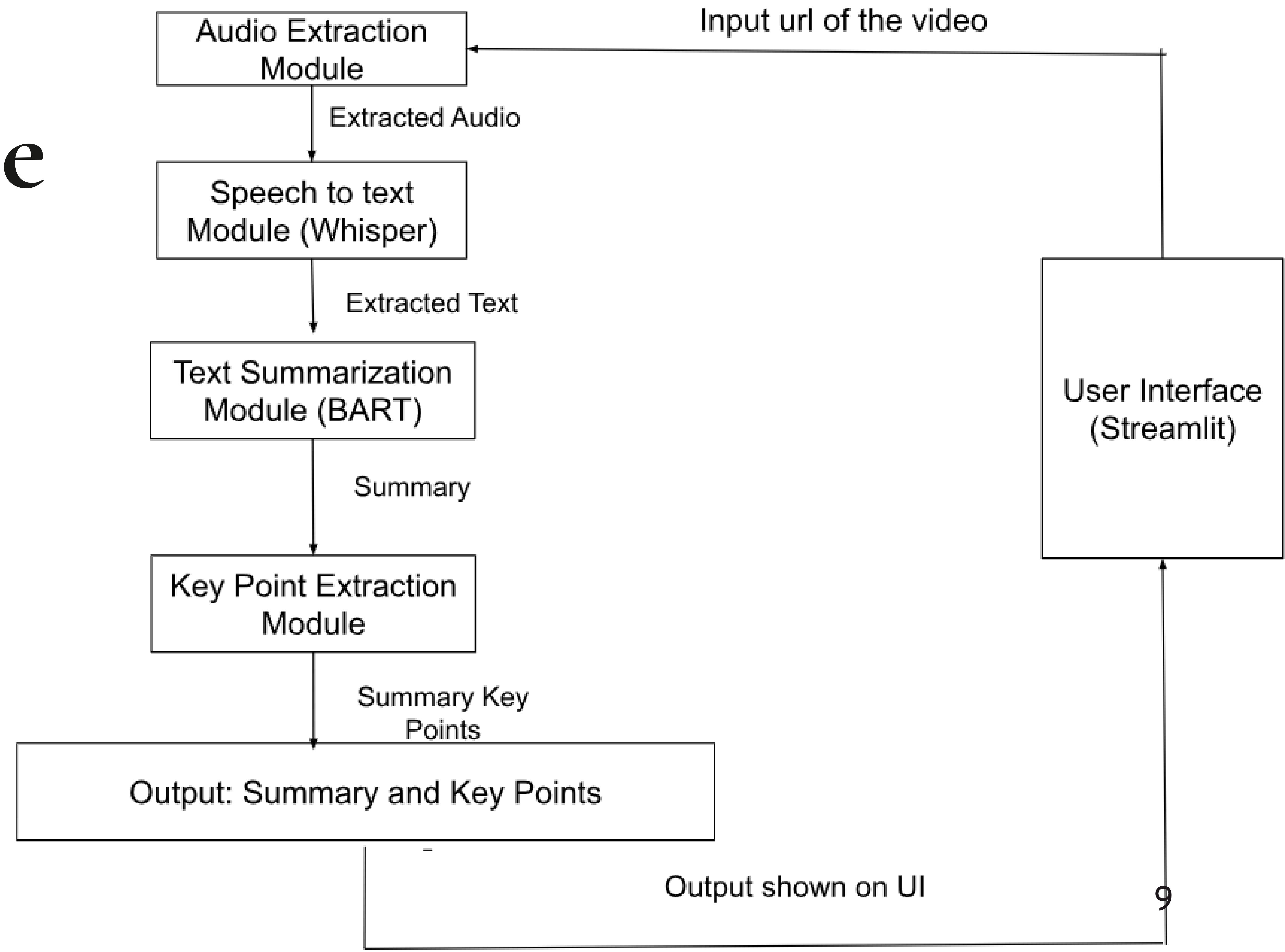
Graphic Card: NVIDIA RTX 3050

GPU: 6 GB and more

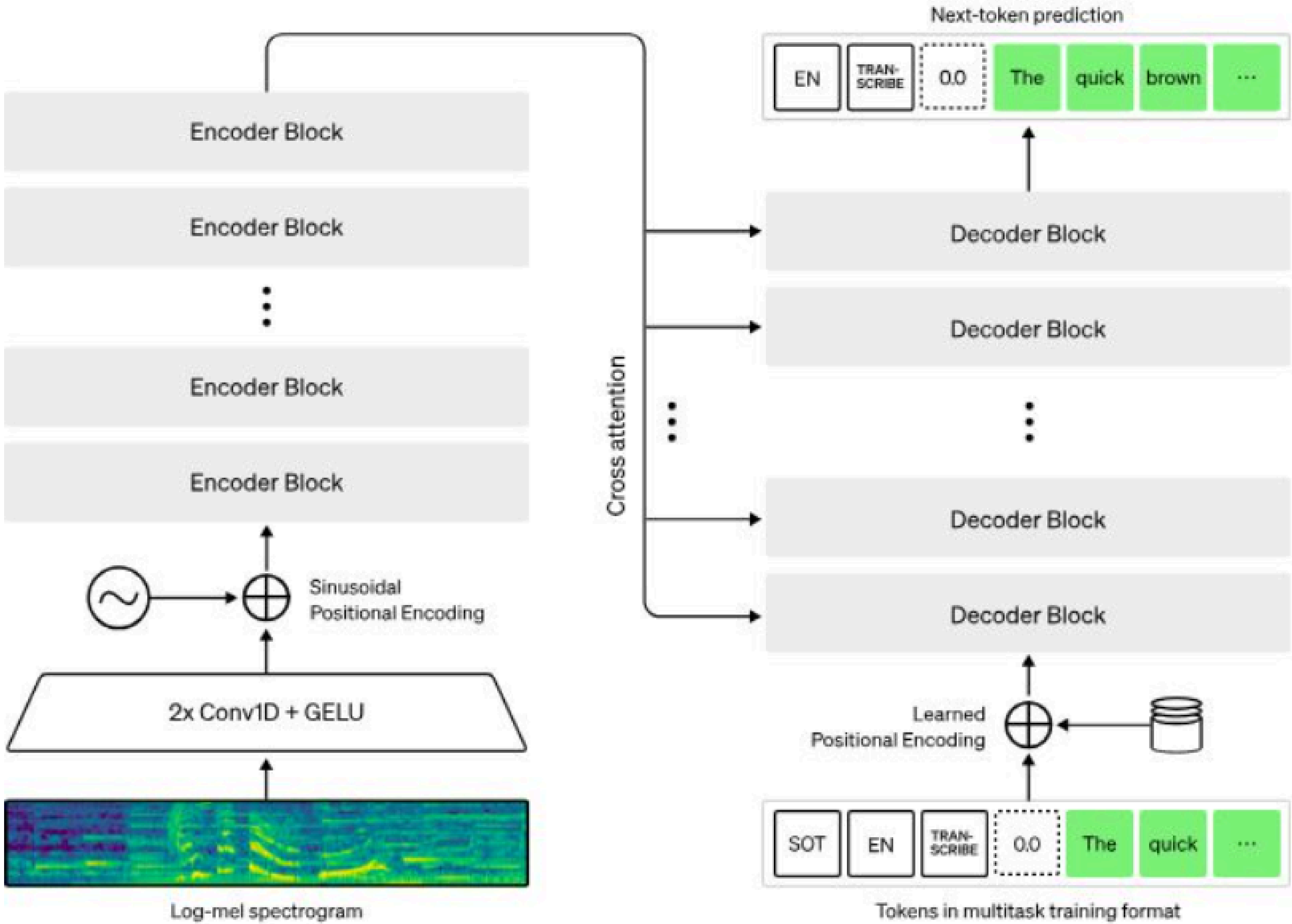
CPU: 64-bit operating processing system with 2.40GHz

Methodology

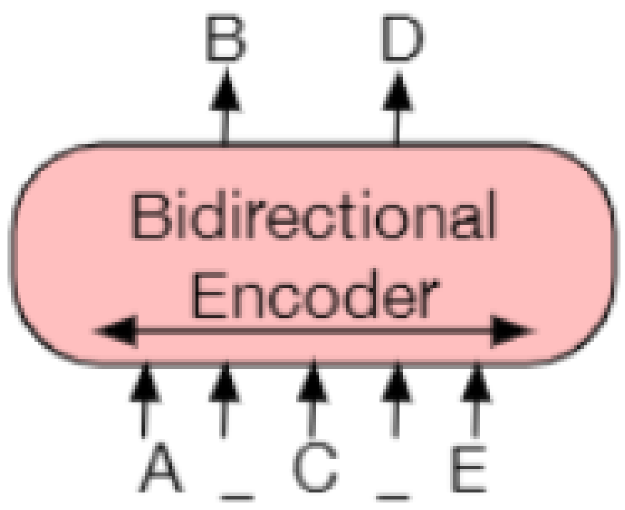
System Architecture


Input url of the video
Audio Extraction Module
Extracted Audio
Speech to text Module (Whisper)
Extracted Text
Text Summarization Module (BART)
Summary
Key Point Extraction Module
Summary Key Points
Output: Summary and Key Points
Output shown on UI
User Interface (Streamlit)

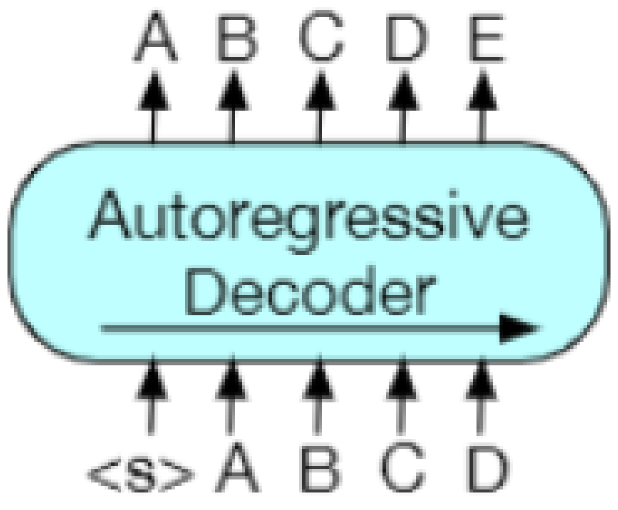
Whisper Model Architecture



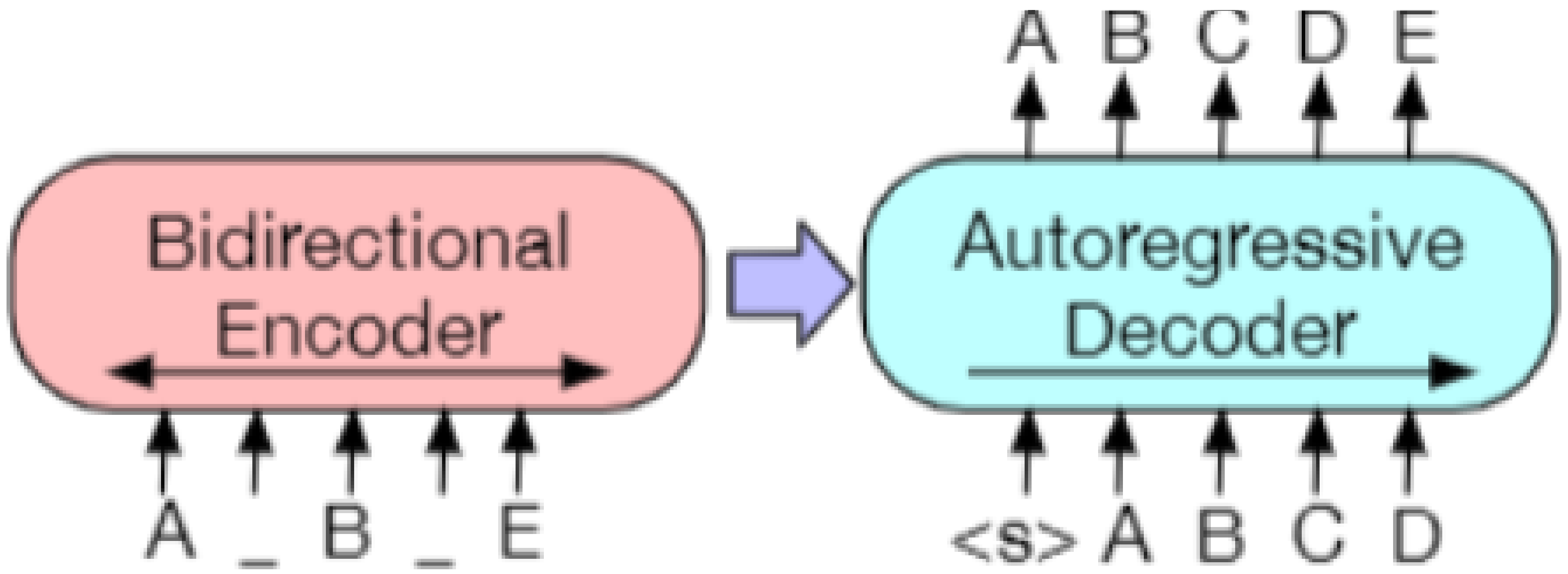
BART Model Architecture



a. BERT



b. GPT



c. BART

Implementation

Audio Extraction Module

Input Sources:

- The system accepts both online video sources (e.g., YouTube URLs) and local video files.

Technology Used:

- yt_dlp library for downloading audio from online videos.
- ffmpeg for extracting audio tracks from local video files.

Process Flow:

- The system first detects whether the input is a URL or a file path.
- Based on the input type, the appropriate method is called to extract the audio and save it in a suitable format for transcription.

Speech-to-Text Module

Model Selection:

- The Whisper model by OpenAI is chosen for its high accuracy in transcribing spoken language into text, with support for multiple languages and translation.

GPU Utilization:

- The model is deployed on a GPU to enhance the speed and efficiency of the transcription process, making it suitable for handling large audio files.

Translation Support:

- The system is capable of translating the transcribed text into English if the original audio is in a different language, broadening the system's applicability.

Text Summarization Module

Model Selection:

- The BART model is used for its robust performance in text summarization, leveraging its transformer-based architecture.

Chunking Mechanism:

- To manage long transcriptions, the text is divided into smaller chunks based on token limits, ensuring that the model processes the text efficiently.

Summary Generation:

- Each chunk is summarized independently, and the individual summaries are then combined to form a coherent overall summary of the video content.

Key Points Extraction:

- The system extracts key points from the summary by analyzing sentence structures, helping users quickly identify the most critical information.

User Interface Module

Streamlit Integration:

- The system is wrapped in a user-friendly web interface built with Streamlit, allowing users to interact with the system easily.

Input Handling:

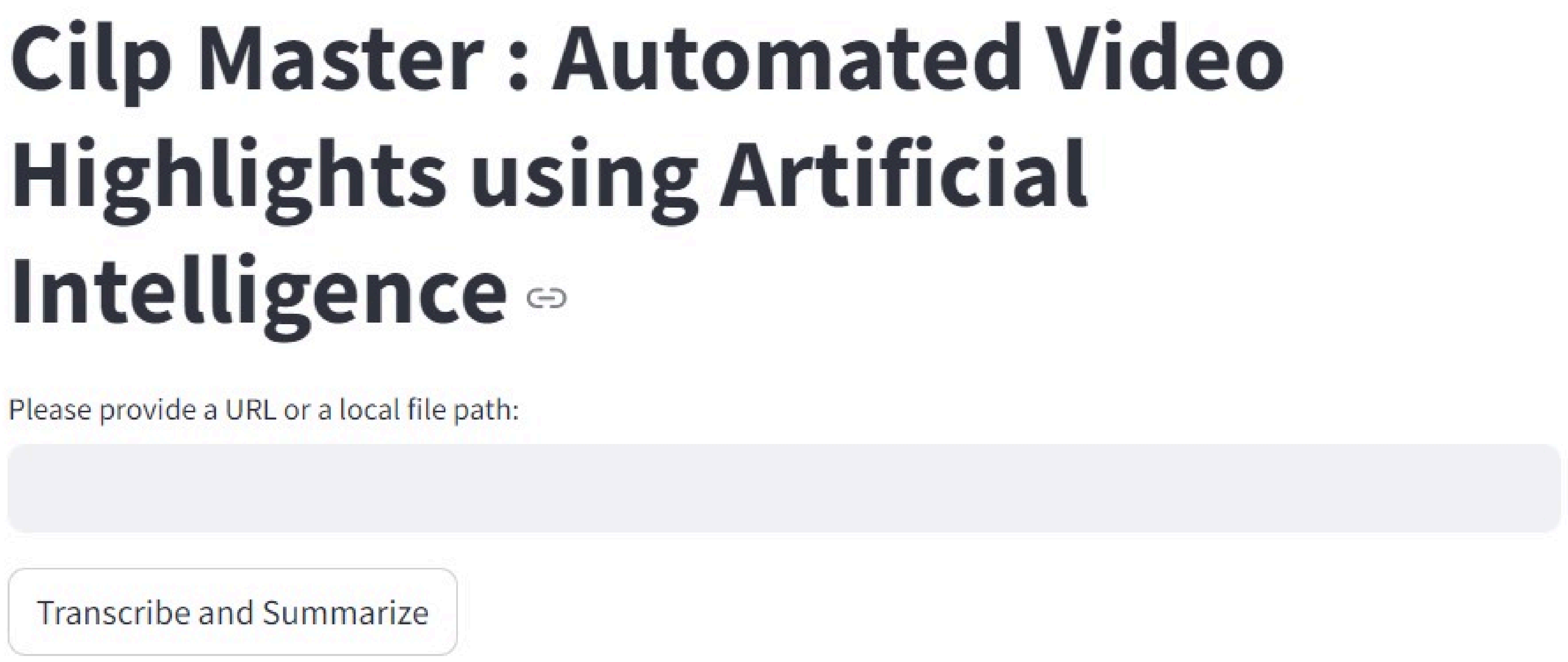
- Users can input a YouTube URL or upload a local video file through the interface.
- A single button click initiates the entire process, from audio extraction to summarization, providing a seamless user experience.

Output Display:

- The summarized text and key points are displayed in an organized manner, enabling users to review the content effectively.

Outputs

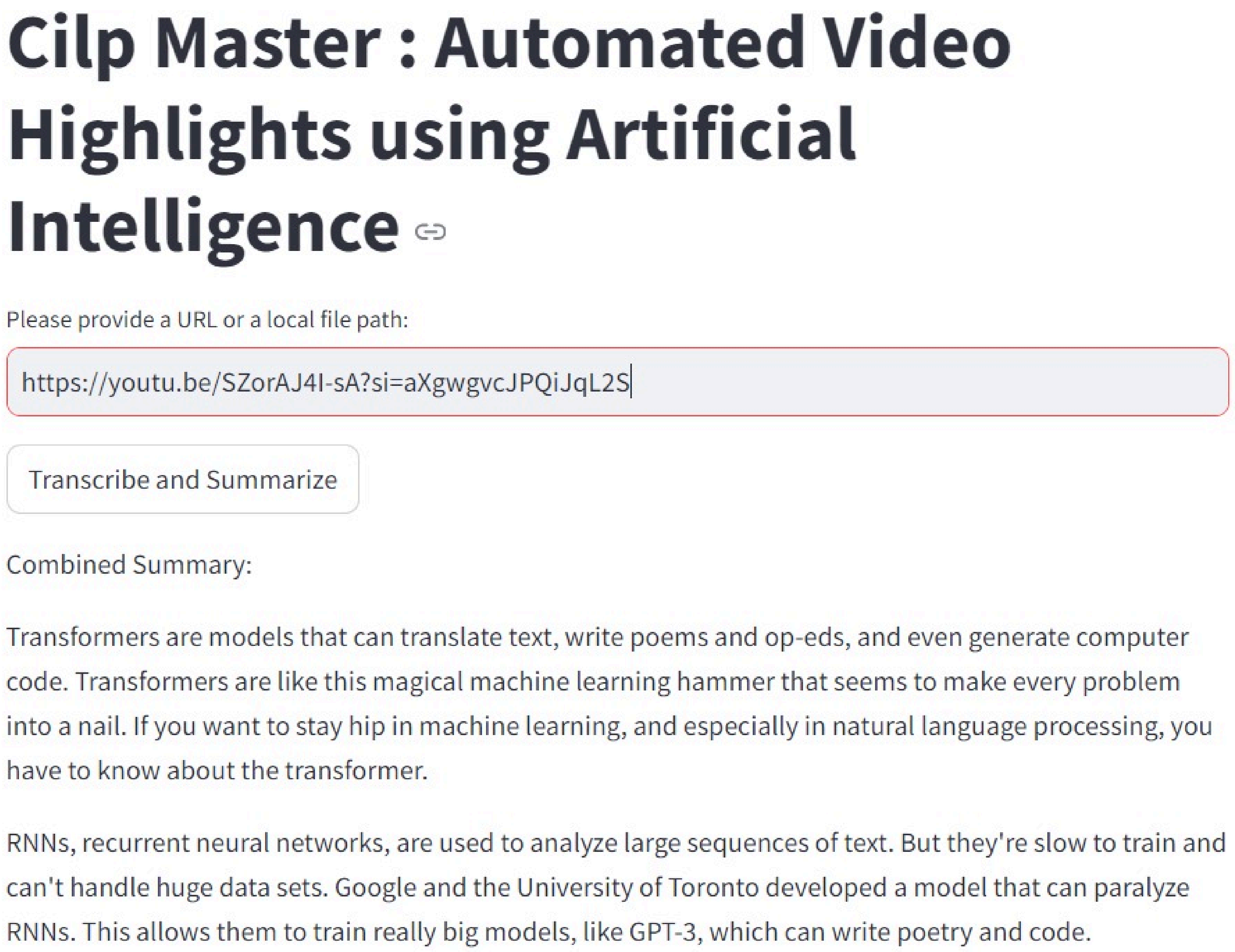
User can provide a link or path of the video.



Cilp Master : Automated Video Highlights using Artificial Intelligence

Please provide a URL or a local file path:

A summary of the video is displayed.



Cilp Master : Automated Video Highlights using Artificial Intelligence

Please provide a URL or a local file path:

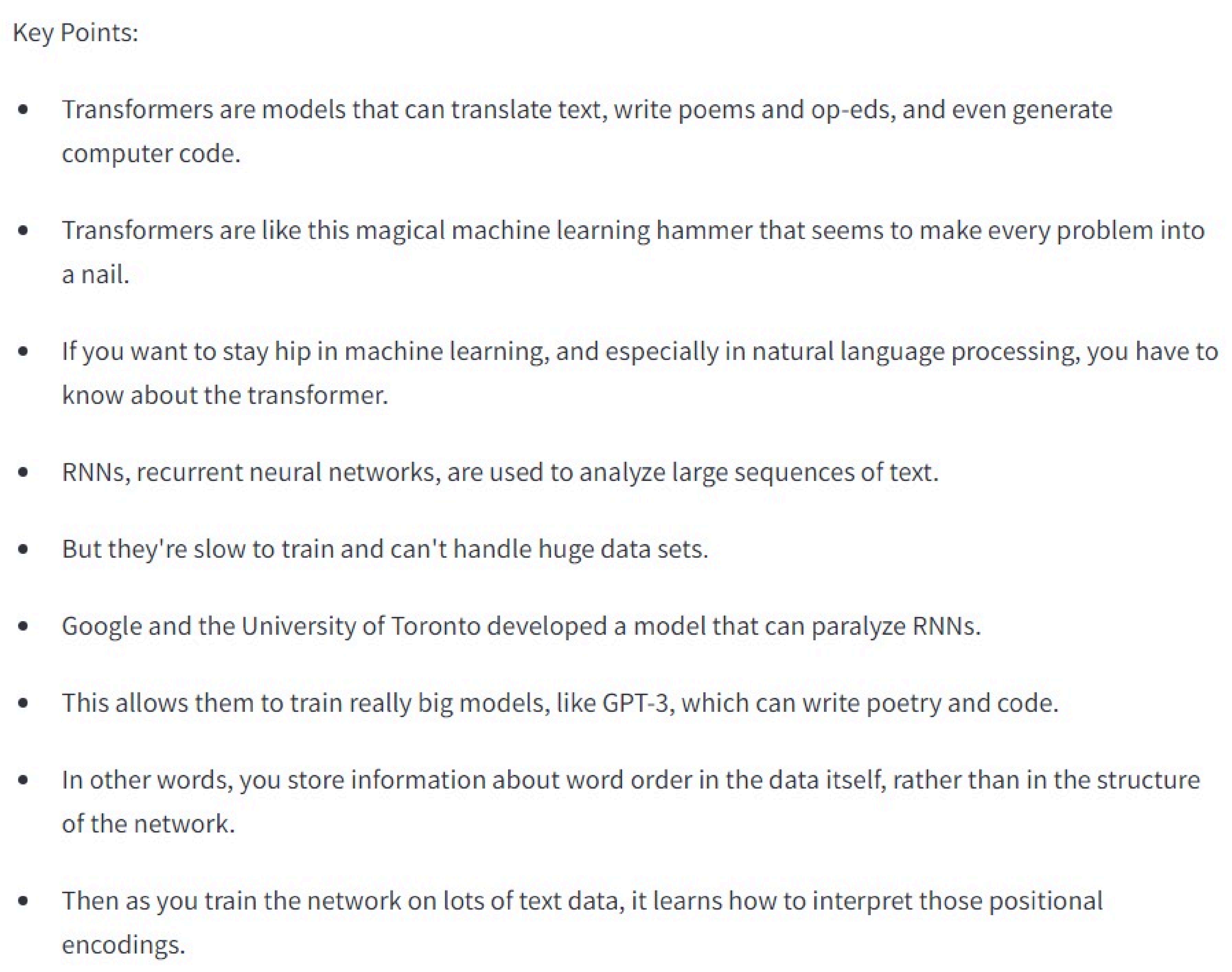
https://youtu.be/SZorAJ4I-sA?si=aXgwgvcJPQiJqL2S

Combined Summary:

Transformers are models that can translate text, write poems and op-eds, and even generate computer code. Transformers are like this magical machine learning hammer that seems to make every problem into a nail. If you want to stay hip in machine learning, and especially in natural language processing, you have to know about the transformer.

RNNs, recurrent neural networks, are used to analyze large sequences of text. But they're slow to train and can't handle huge data sets. Google and the University of Toronto developed a model that can paralyze RNNs. This allows them to train really big models, like GPT-3, which can write poetry and code.

Key highlights of the summary are displayed.



Key Points:

- Transformers are models that can translate text, write poems and op-eds, and even generate computer code.
- Transformers are like this magical machine learning hammer that seems to make every problem into a nail.
- If you want to stay hip in machine learning, and especially in natural language processing, you have to know about the transformer.
- RNNs, recurrent neural networks, are used to analyze large sequences of text.
- But they're slow to train and can't handle huge data sets.
- Google and the University of Toronto developed a model that can paralyze RNNs.
- This allows them to train really big models, like GPT-3, which can write poetry and code.
- In other words, you store information about word order in the data itself, rather than in the structure of the network.
- Then as you train the network on lots of text data, it learns how to interpret those positional encodings.

Conclusion

Conclusion

- The project successfully demonstrates an efficient and scalable system for video-to-text transcription and summarization by integrating state-of-the-art technologies such as OpenAI's Whisper model and Facebook's BART model.
- The combination of these tools with Streamlit for an intuitive user interface allows for seamless extraction, transcription, and summarization of audio from video content, addressing the growing demand for accessible and summarized media.
- The use of ffmpeg for audio extraction, yt-dlp for video downloading, and advanced NLP techniques for summarization ensures that the system is both robust and versatile.
- This project not only highlights the potential of AI in automating complex tasks but also opens up opportunities for further enhancements, such as real-time processing and multilingual support.

Future Scope

- Real-Time Transcription and Summarization
- Multilingual Support
- Integration with Content Management Systems (CMS)
- User-Customizable Summaries
- Sentiment Analysis and Key Insights Extraction
- Mobile Application Development
- Chatbot Integration

References

- Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2021). *Robust Speech Recognition via Large-Scale Weak Supervision*. OpenAI. Available at: https://github.com/openai/whisper
- Streamlit Inc. (2022). *Streamlit: The fastest way to build and share data apps.* Available at: https://streamlit.io/
- yt-dlp Developers. (2021). *yt-dlp: A youtube-dl fork with additional features and fixes.* Available at: https://github.com/yt-dlp/yt-dlp
- FFmpeg Developers. (2022). *FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video*. Available at: https://ffmpeg.org/

Thank You