YouClipAI - Automating YouTube Video Data Collection with Large Language Models

Mu-Ruei Tseng
Texas A&M University
133007868

1 Problem Statement

Scraping video content from platforms like YouTube presents a significant challenge for efficient data collection due to limitations in its search algorithms. YouTube primarily indexes videos based on titles, tags, and descriptions, which often fails to capture the specific content users are looking for. More importantly, this approach makes it difficult to locate the exact timestamps of the desired content within a video. Users are left to manually browse through videos to find the relevant sections, a time-consuming and inefficient process, particularly when collecting in-the-wild video clips for research or analysis.

This project aims to address this challenge by developing a tool that leverages Large Language Models (LLMs) to automatically extract relevant video clips based on user prompts. Instead of relying on metadata, the tool will use LLMs to understand the actual content of the video, accurately identifying and extracting the precise start times of segments that match the user's description. By automating this process, we can significantly reduce the time and effort involved in video search and data collection. The project not only improves the precision of video searches but also enables more efficient research workflows that rely on accurate and relevant video data.

2 Project Objectives

The primary goal of this project is to create a tool that automatically extracts relevant video clips from platforms like YouTube based on user prompts. For instance, if a user requests a 10-second clip of Obama giving a speech, the tool will return multiple matching clips along with their precise timestamps. Leveraging Large Language Models (LLMs), the tool will analyze video content to provide accurate results.

An essential objective is integrating this tool as a Chrome extension, enabling users to easily search for and retrieve specific video segments directly from their browser. Additionally, a version utilizing local LLMs will be developed to address potential costs associated with public API keys to offer a cost-effective solution.

The expected outcome is significantly reducing the time and effort required for video searches, enhancing precision, and improving workflow efficiency, particularly for users engaged in research and data collection tasks.

3 Methodology

3.1 Model Architecture

The proposed architecture integrates two main components: candidate selection and local content selection, each designed to optimize the relevance and accuracy of extracted video segments based on user prompts. See Figure 1.

Candidate Selection This initial phase employs a large language model to interpret user inputs and generate structured queries following the 4W framework—What, Where, When, and Who. These queries drive a targeted search on YouTube, facilitated by predefined hyperparameters such as T (maximum video duration) and N (maximum number of search results), which help streamline the initial screening of content. Additionally, a video summarization model can be applied to further refine the search results, ensuring they align closely with the user's specified criteria. This methodical approach significantly enhances the precision in selecting potential video candidates.

Local Content Selection Once candidates are selected, the local content selection process begins, focusing on the detailed analysis of each video. Audio content from the videos is processed using automatic speech recognition (ASR) models to transcribe spoken words into text, allowing for a script-based search. This script is then analyzed with an LLM to identify segments where the script's content matches the 'What' aspect of the user's prompt. Recognizing the limitations of audio information in confirming other critical details like 'Who' and 'Where', the system incorporates an object recognition model to analyze visual frames within these identified segments. This dual analysis of audio and visual data ensures comprehensive verification and selection of content that precisely matches the user's intent.

This robust, two-tiered architecture facilitates a thorough and efficient method for extracting targeted video content, capitalizing on the synergistic use of LLMs, ASR, and object recognition technologies to deliver highly relevant and accurate results.

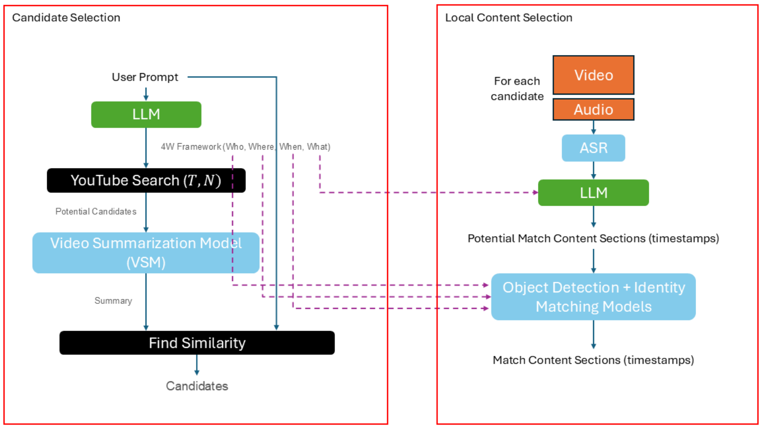
Candidate Selection
User Prompt
LLM
4W Framework (Who, Where, When, What)
YouTube Search (T, N)
Potential Candidates
Video Summarization Model (VSM)
Summary
Find Similarity
Candidates
Local Content Selection
For each candidate
Video
Audio
ASR
LLM
Potential Match Content Sections (timestamps)
Object Detection + Identity Matching Models
Match Content Sections (timestamps)

Figure 1: Overview of the YouClipAI model.

3.2 LLM(s) and Techniques

A variety of Large Language Models (LLMs) provide accessible APIs for integration, such as OpenAI's GPT and the Gemini API. Although these APIs offer ease of use, they typically incur costs per query. An alternative is deploying a local instance of Llama 3, which could reduce operational costs while serving as the project's primary LLM. For Automatic Speech Recognition (ASR), models like OpenAI's Whisper (6) or AUTO-AVSR (5) are recommended due to their robust performance. Identity matching can be effectively handled by ArcFace (1), a well-known model for facial recognition tasks. For video summarization, Shotluck-Holmes (4) is an excellent choice, known for its precision in detecting scene changes and summarizing video content.

3.3 Data and Evaluation

Data for this project will be sourced from YouTube, where numerous videos include manually created timestamps in the descriptions or comments. These timestamps, which often highlight specific content within the videos, will be scraped and utilized as evaluation samples. This approach leverages existing annotated data to assess the effectiveness of our video segment extraction methodology systematically.

4 Related Work

In recent years, various methods have been proposed to enhance the efficiency of video content retrieval, particularly through summarization and segmentation techniques. PodSumm(7) employs a two-step model that first transcribes audio into text using an Automatic Speech Recognition (ASR) model, followed by a BERT-based architecture for text summarization to generate concise summaries from podcast transcripts. On the other hand, He et al.(2) introduced a transformer-based model for multimodal summarization that combines both video and text inputs to leverage cross-modal information and improve summary quality. While these approaches offer valuable insights for summarizing general content, they do not address the specific challenge of pinpointing detailed content within videos based on user queries.

Lin et al. (3) focused on a linguistic-based approach to segment lengthy lecture videos. Their method uses traditional natural language processing techniques such as noun phrase extraction to identify features from a given text and perform matching accordingly. However, this approach only considers audio information while ignoring video content, and feature matching alone may be outdated due to the introduction of more advanced techniques such as Large Language Models.

This project, unlike prior methods that focus on broad summarization or segmentation, is designed to precisely locate and extract specific moments within videos harnessing both audio and image information given specific user inputs. This addresses the challenge of efficiently retrieving exact video timestamps. Additionally, by integrating this functionality into a user-friendly browser extension, we provide an accessible, scalable solution for real-time video content retrieval.

5 Timeline

5.1 Phase 1 (First month)

- Objective: Collection of models and implementation of the candidate selection component.
- Milestones:
  - Model Collection: Gather public models such as Large Language Models (LLMs), Automatic Speech Recognition (ASR), Identity Matching models, and video summarization tools that are relevant to the project. (First one to one and a half weeks.)
  - Data Collection: Acquire datasets necessary for model evaluation. (First one to one and a half weeks.)
  - Implementation of Candidate Selection: Develop and integrate the candidate selection section. (Second week to the end of the month.)

5.2 Phase 2 (Second month)

- Objective: Develop the local content selection and prepare the user API for release.
- Milestones:
  - Implementation of Local Content Selection: Finalize the audio and video content matching modules. (First week of the second month to the third week.)
  - API Development and Testing: Complete API development and start system-wide testing. (Last week before submission.)
  - Final Code Submission and Report: Submit the final version of the project by the end of the second month. (Last week before submission.)

6 Challenges and Risks

Implementing the proposed architecture can present several challenges and risks:

1. Handling Long Videos: Long videos are complex and require intensive processing. Summarization models may struggle with accuracy over extended durations, risking the omission or misinterpretation of key details.

2. Accuracy of ASR and Object Recognition Models: Variability in the accuracy of Automatic Speech Recognition and object recognition systems is notable, especially with poor audio quality, multiple speakers, diverse accents, or complex scenes.
3. Dependency on LLM Precision: The effectiveness of Large Language Models in generating and refining queries heavily depends on their training and the clarity of user prompts. Ambiguities in prompts or limitations in the LLM's understanding can result in less targeted searches, leading to inefficient retrieval of relevant content.

7 Expected Deliverables

The repository for this project, including the code, final report, and demo videos, is available at https://github.com/Morris88826/YouClipAI. If you would like access to the repository or have any questions, please feel free to contact me at mtseng@tamu.edu.

References

[1] Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)

[2] He, B., Wang, J., Qiu, J., Bui, T., Shrivastava, A., Wang, Z.: Align and attend: Multimodal summarization with dual contrastive losses (2023), `https://arxiv.org/abs/2303.07284`

[3] Lin, M., Chau, M., Cao, J., Jr, J.: Automated video segmentation for lecture videos: A linguistics-based approach. IJTHI **1**, 27–45 (01 2005)

[4] Luo, R., Peng, A., Vasudev, A., Jain, R.: Shotluck holmes: A family of efficient small-scale large language vision models for video captioning and summarization (2024), `https://arxiv.org/abs/2405.20648`

[5] Ma, P., Haliassos, A., Fernandez-Lopez, A., Chen, H., Petridis, S., Pantic, M.: Auto-avsr: Audio-visual speech recognition with automatic labels. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE (Jun 2023). https://doi.org/10.1109/icassp49357.2023.10096889, `http://dx.doi.org/10.1109/ICASSP49357.2023.10096889`

[6] Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision (2022), `https://arxiv.org/abs/2212.04356`

[7] Vartakavi, A., Garg, A.: Podsumm – podcast audio summarization (2020), `https://arxiv.org/abs/2009.10315`