YouClipAI - Automating YouTube Video Data Collection with Large Language Models

Mu-Ruei Tseng
Texas A&M University
133007868

Abstract

The rapid growth of online video content has significantly transformed how people access information. Platforms like YouTube host an extensive range of videos across diverse categories, offering users a wealth of resources. However, while this vast collection is undeniably valuable, it presents notable challenges. Locating specific information within a video can be a tedious and time-consuming task, as the platform's search functionality largely depends on metadata such as titles, tags, and descriptions. Consequently, users often find themselves manually navigating through lengthy videos to pinpoint the exact moments of interest. This inefficiency has highlighted the need for a smarter solution—one that understands the content within a video rather than relying solely on descriptive metadata. To address this issue, we introduce YouClipAI, a platform that integrates video search with Large Language Models (LLMs), offering a precise, content-driven method for identifying and extracting specific video clips that best match the user's query.

Keywords Video Scraping, Large Language Models, Automatic Speech Recognition

1 Introduction

In today's digital age, video content plays a central role in entertainment, education, information sharing, and research. Despite its importance, finding specific moments in the vast collection of online videos remains a major challenge. Platforms like YouTube rely heavily on metadata such as titles, tags, and descriptions to organize and search videos. However, without timestamps provided by the video creators, locating a specific clip within a long video can be difficult. This often forces users to manually search through the content, which is time-consuming and inefficient. For example, if someone wants to find a moment in a press conference where a basketball player talks about playing with his son in the league, they might need to skim through the entire video manually, making the task tedious.

The development of Large Language Models (LLMs), which excel at understanding text and context, offers new possibilities for solving this problem. LLMs can analyze video transcripts, captions, and related metadata to find specific moments based on natural language queries. Unlike traditional search methods that rely on metadata or timestamps, LLMs use advanced language understanding to locate the relevant parts of a video more effectively.

However, for LLMs to work well, they need accurate transcripts of the video content. Automatic Speech Recognition (ASR) technology is crucial in this process, as it converts spoken language in videos into text. By combining ASR with LLMs, it is possible to automate the analysis of video content, making it easier to find specific clips even when no pre-generated transcripts or captions are available.

The goal of this project is to introduce YouClipAI, a tool that simplifies finding and extracting important moments from video content. This tool reduces the need for manual searching and

improves the accuracy of video retrieval by integrating advanced technologies such as Whisper ASR for transcription and LLM for content extraction.

YouClipAI is designed to help a wide range of users: researchers can gather video clips for analysis, journalists can quickly find quotes or scenes for stories, and content creators can locate key moments without spending hours watching videos. By automating the process of video content extraction, YouClipAI streamlines workflows in research, content creation, and analysis, allowing users to focus on what truly matters.

2 Related Work

Various methods have been proposed to improve the efficiency of video content retrieval, with a primary focus on summarization and segmentation techniques. For instance, PodSumm(4) employs a two-step model that transcribes audio into text using an Automatic Speech Recognition (ASR) model, followed by a BERT-based architecture for text summarization, creating concise summaries from podcast transcripts. Similarly, He et al.(1) introduced a transformer-based model for multimodal summarization, combining video and text inputs to leverage cross-modal information and enhance summary quality. While these approaches demonstrate advancements in content summarization, they do not address the specific challenge of locating detailed content within videos based on user queries.

In another approach, Lin et al. (2) proposed a linguistic-based method for segmenting lengthy lecture videos. Their method uses traditional natural language processing techniques, such as noun phrase extraction, to identify textual features and perform feature matching. However, this method relies solely on audio data, overlooking valuable video content, and is limited by the outdated nature of traditional feature-matching techniques compared to the capabilities of modern Large Language Models (LLMs).

More recent works have explored the integration of ASR and LLMs for video content analysis. For example, tools like Whisper (3) have demonstrated high accuracy in transcribing spoken language into text, providing a strong foundation for further analysis. Similarly, advances in LLMs such as GPT-based architectures have enabled semantic search and query-based retrieval from textual data, which can be extended to video transcripts for pinpointing specific moments.

This project builds upon these advancements but addresses a gap in existing methods: the precise retrieval of specific video segments based on detailed user queries. Unlike broad summarization or segmentation approaches, our method focuses on locating the exact timestamps of interest.

3 Methodology

In YouClipAI, we offer two types of search modes: Quick Start Mode and Advanced Search Mode. In Quick Start Mode, users simply provide a YouTube URL containing the video they believe holds the desired information, along with a query to extract the relevant segment. Advanced Search Mode, on the other hand, allows users to input a broader prompt, enabling the system to search across relevant YouTube videos and return the most suitable clips based on the query.

3.1 Quick Start

Figure 1 illustrates the UI for the Quick Start page. The workflow for this page is straightforward: the user inputs the desired YouTube URL and clicks Fetch Video to extract metadata (e.g., title, duration). After fetching the video information, clicking Analyze Video triggers the Whisper ASR to transcribe the audio. I use the Turbo model, which has a Word Error Rate (WER) of approximately 7.75% on English Automatic Speech Recognition datasets.

To handle long videos, the system divides them into smaller 2-minute segments with a 50% overlapping window to ensure seamless processing of content spanning across segments. Accurate timestamps are critical for our application, as they allow precise start and end periods for extracted information. Therefore, each word in the transcript is saved as a token along with its corresponding start and end time (in seconds). See Figure 2 for more details.

With the per-word transcript and corresponding timestamps prepared, the next step is integrating it with the LLM model. For this project, I use GPT-4o in conjunction with LangChain, with a maximum

token limit set to 512. The LLM processes the user's query by extracting relevant information using the 4W1H format (Who, What, When, Where, and How). The 4W1H format is employed because it provides a comprehensive structure for extracting key details from unstructured data. This approach ensures that the extracted information is clear, actionable, and relevant to the user's query. By focusing on these fundamental aspects, the system can deliver precise and contextually meaningful results that align with user needs. The engineered prompt is shown in the Appendix 7.1.1.

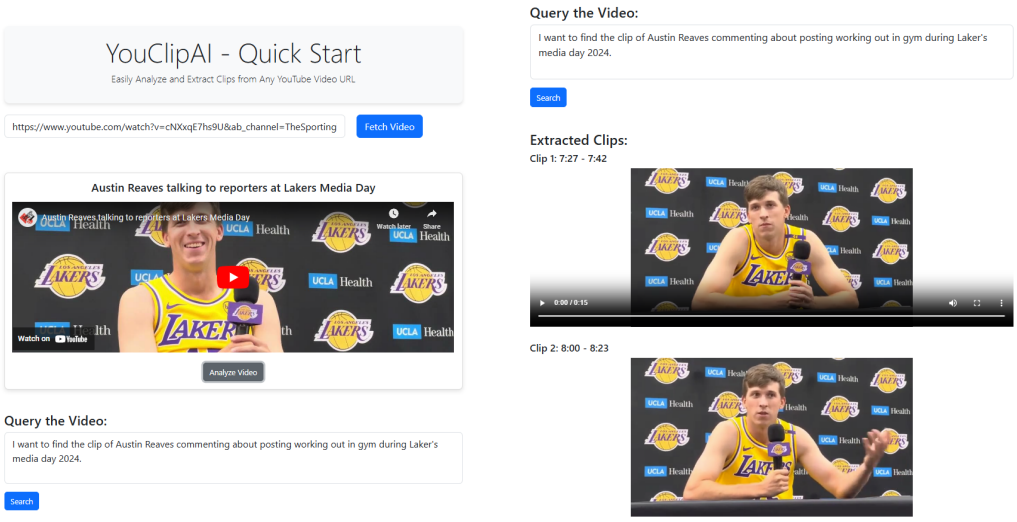

Figure 1: Quick Start Page - Provide YouTube URL and Query to Extract Relevant Segment

To locate the specific clip that aligns with the user's query based on the 4W1H framework, I designed a prompt (Appendix 7.1.2) that combines both the user's query and the relevant transcript information. The prompt explicitly instructs the model to return "None" if no matching content is found, ensuring accurate and meaningful results. This allows the LLM to look into each transcript and extract possible sections that meet the query requirements.

After extracting potential clip segments, the model further analyzes their content against the original query using the LLM and ranks them based on similarity. The prompt is designed to ensure that overlapping or closely related segments are merged based on their timestamps and content, preventing the return of multiple short, redundant sections. The detailed prompt can be found in 7.1.3. Finally, the extracted clip will be saved in the download folder and displayed on the front end for the user to view.

3.2 Advanced Search

On this page, users are relieved from manually inputting the YouTube URL and are provided with a text field to specify the video clip they want to find. The overview pipeline is shown in Figure 3. The 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.

```
word,start,end
 is,120.0,120.18
 going,120.18,120.32
 to,120.32,120.38
 be,120.38,120.6
 what,120.6,121.16
 people,121.16,121.42
" like,",121.42,121.66
```

Figure 2: Sample Transcript

Candidate Selection Similar to the previous section, this initial phase utilizes a large language model (LLM) to interpret user inputs and generate structured queries based on the 4W1H framework—What, Where, When, Who, and How. The LLM is then used to refine the query specifically for web scraping on YouTube. For scraping YouTube videos, I use Selenium to simulate a YouTube search. While I initially experimented with the YouTube API, the results were suboptimal in terms of relevance and quality. As an alternative, I scrape the top 20 videos from the search results and use the LLM again, in combination with the original query, to evaluate and rank the similarity between the user query and the titles of the retrieved videos.

Local Content Selection Once the candidate videos are selected, the local content selection process begins, as previously described, focusing on a detailed analysis of each video. The audio content is processed using an automatic speech recognition (ASR) model to transcribe spoken words into text, enabling a transcript-based search. The transcript is then analyzed with a large language model (LLM) to identify segments where the content aligns with the user's prompt. Subsequently, the LLM is used to merge overlapping segments and rank the extracted clips based on their relevance.

This two-tiered architecture provides a robust and efficient approach for extracting targeted video content, leveraging the combined strengths of LLMs and ASR technologies to deliver precise and contextually accurate results. The UI is shown in Figure 4.

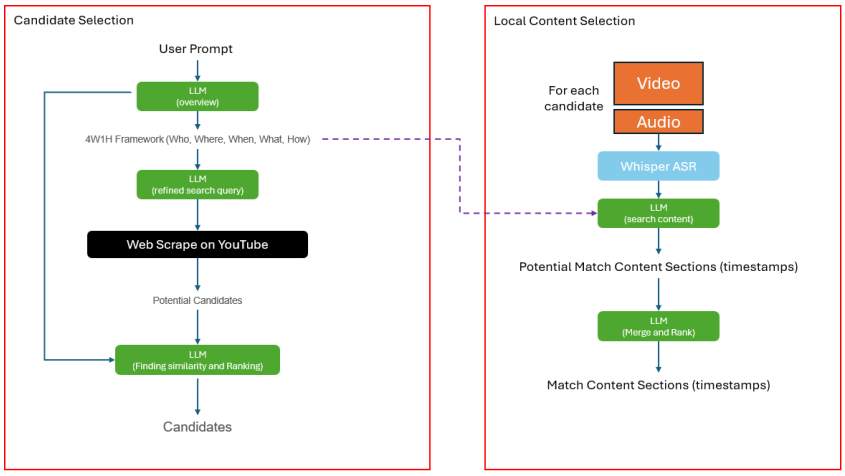

Figure 3: Overview of the Advanced Search in YouClipAI.

4 Results

In this section, I will show several queries and extracted clip pairs from the YouClipAI.

4.1 Result 1

- Query: What is Elon Musk's idea on AI?
- Extracted Clip 1 (6:06 - 6:44): He really seemed to be one sort of digital superintelligence, basically digital god, if you will, as soon as possible. He wanted that? Yes. He's made many public statements over the years that the whole goal of Google is what's called AGI, artificial general intelligence or artificial superintelligence. And I agree with him that there's great potential for good, but there's also potential for bad. And so if you've got some radical new technology, you want to try to take the set of actions that maximize probably it will do good and minimize probably it will do bad things.
- Extracted Clip 2 (10:25 - 10:50): As a human being, it's okay to look out for human beings first. And then at the end, he said the real problem with A.I. is not simply that it will jump the boundaries and become autonomous and you can't turn it off. In the short term, the problem with A.I. is that it might control your brain through words. And this is the application that we need to worry about now, particularly going into the next presidential election.

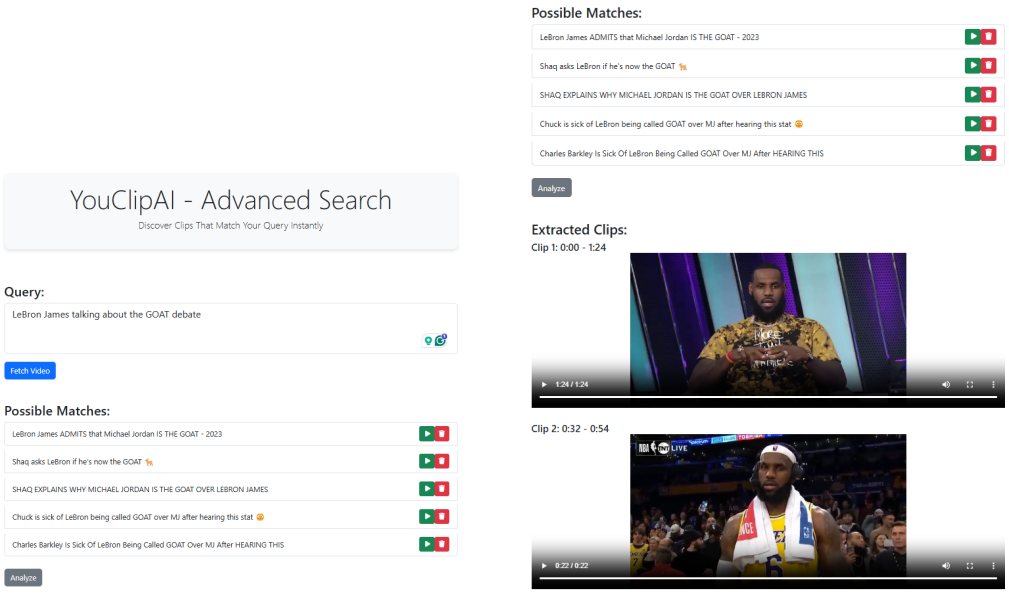

Figure 4: Advanced Search Page: Enter a Query to Find Relevant Video Clips

Figure 5: Result 1: What is Elon Musk's idea on AI?

4.2 Result 2

For this sample, we observe that extracted clips 1 and 2 align with the content provided. However, clip 3, while discussing the GOAT debate, is not spoken by LeBron. This highlights the limitation of relying solely on the transcript for analysis. Without incorporating visual or audio cues, the system cannot confirm whether the desired subject is the speaker in the identified clip.

- Query: LeBron James talking about the GOAT debate
- Extracted Clip 1 (0:00 - 1:24): And you were like, you know what? I'm smarter. I'm wiser even than I was four years ago. You know what? That's not for me to decide. That's for everybody else. People are always going to argue that. So, like, I view it as a moment of maturity, right? Like, why is it that you took a step back in that moment? Not saying that you changed your mind or any of that. Because I realize that the conversation is going to continue to happen no matter what. And it's great for the game. Yeah. It's great for the barbershops. It's great for the game. It's going to continue to happen. I don't know why. The one thing I don't understand is, like, you know, you never hear, you know, the greatest quarterbacks. Yeah. Always talk about it. You never hear, like, who's the greatest quarterback, you know, Brady or Montana. You know, you got Montana with five. You got Brady with six. Yeah. You know, or, you know, you used to hear the Peyton and Brady thing when they played. But, you know, Brady started to win a lot more and things of that

nature. But you never hear, like, every single conversation. Baseball. Baseball. It's like Babe Ruth and then people just move on. Right. They just move on. No more arguments. Babe Ruth is the greatest baseball player. Right. Exactly. You know, you know, so, you know, and other sports, hockey, you hear Wayne Gretzky. That's it. You know, people move on. You haven't heard anything since. Yeah. You move on. Wayne Gretzky is the greatest. You move on. You know, things of that nature. But, you know, the conversation is going to always be had. And it's great for debates because they want to compare errors. They want to compare players and

- Extracted Clip 2 (0:32 - 0:54): I'm gonna take myself. I'm gonna take myself against anybody that's ever played this game. You know but everyone's gonna have their favorite. Everyone's gonna you know decide who their favorite is. But I know what I've brought to the table. I know what I bring to the table every single night and what I can what I can do out on this floor. So you know I always feel like I'm the best to ever play this game. But you know there's so many other great ones and I'm happy to just be a part of their part of their journey.
- Extracted Clip 3 (1:09 - 1:23): So LeBron has played how many more seasons than Michael Jordan? He's still behind him? That's crazy. That's crazy. Now, listen, I love LeBron. But for him to be that far behind MJ and I'm playing probably eight more seasons? Come on, man. Y'all need to stop this.

More results and details can be found in the demo video.

5 Challenges and Future Improvements

The current approach encounters several challenges:

1. Handling Long Videos: Processing long videos is computationally intensive and requires significant resources. Currently, all steps, including transcript analysis, are executed sequentially, which increases processing time. Additionally, the system is restricted to handling videos of up to 20 minutes, limiting its applicability for longer content.
2. Deployment Challenges: Deploying the system online presents difficulties due to the need to download videos for analysis, which requires substantial storage and bandwidth. The current workaround involves running the backend locally, but a fully online deployment solution remains under exploration.
3. Integration with Visual Content: At present, the analysis relies solely on transcripts, which are influenced by the accuracy of the ASR model. Furthermore, transcripts do not capture all the information contained within video content. Future development could incorporate vision models, such as object recognition or image captioning, to enhance the robustness and comprehensiveness of the extracted clips.

6 Code and Implementation

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.

7 Appendix

7.1 Prompts

7.1.1 Overview Prompt

```
"Analyze the query and extract the relevant information according to the 4W1H framework (Who, What, When, Where, How).\n"
"Query: {query}\n"
"You MUST provide a response for each category in the following format, even if it is blank.\n"
"Respond in this JSON format:\n"
"{{\n"
"  \"Who\": \"...\",\n"
"  \"What\": \"...\",\n"
"  \"When\": \"...\",\n"
"  \"Where\": \"...\",\n"
"  \"How\": \"...\"\n"
"}}"
```

7.1.2 Search Content Prompt

```
"You are an AI assistant specialized in extracting relevant sections from transcripts based on the 'What' information provided in the context of a 4W1H framework (Who, What, When, Where, Why, and How). \n"
"Instructions:"
"1. Identify and extract the section that CONTAINS information related to the 'What' provided.\n"
"2. If no relevant section is found, return 'None' for all fields.\n"
"3. You MUST ensure that the extracted content is the LONGEST continuous section that covers the 'What' information.\n"
"Transcript [(word, start_time, end_time), ...]: {transcript}\n"
"What (Information to extract): {What}\n"

"You MUST return in the following format:\n"
"{{\n"
"  \"content\": \"The relevant section from the transcript or 'None' if no match.\",\n"
"  \"info\": \"Explanation or context of the relevant section or 'None' if no match.\",\n"
"  \"start_time\": \"The start time of the relevant section or 'None' if no match.\",\n"
"  \"end_time\": \"The end time of the relevant section or 'None' if no match.\"\n"
"}}"
```

7.1.3 Merge and Rank Prompt

```
"You are an AI assistant tasked with analyzing, merging, and ranking search results based on their relevance to a given query.\n\n"
"Instructions:\n"
"1. Identify overlapping or closely related sections:\n"
"   - Two sections are considered overlapping if their time ranges intersect or if the end time of one section is within 5 seconds of the start time of another.\n"
"   - Two sections are considered closely related if their content discusses the same topic or has thematic similarity to the query.\n"
"   - Merge sections only if they meet both criteria: overlapping time ranges and thematic similarity.\n"
"2. Rank the merged sections based on their relevance to the query:\n"
"   - Prioritize sections that directly address the query.\n"
"   - Rank higher sections that are more specific, unique, and detailed in their relevance to the query.\n"
"3. The MOST relevant section should be at the start of the list.\n"

"Search Results: {search_results}\n"
"Query: {query}\n\n"

"You MUST return the ranked results in the following format:\n"
"{{\n"
"  \"ranked_results\": [\n"
"    {{\n"
"      \"start_time\": \"Earliest start time of the merged section, MUST be in float\",\n"
"      \"end_time\": \"Latest end time of the merged section, MUST be in float\"\n"
"    }},\n"
"    ...\n"
"  ]\n"
"}}"
```

References

[1] 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

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

[3] 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

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