CS 510 - Project Proposal - Development Track

Co-ordinator - Pranav Gor (gor2@illinois.edu)
Yash Damania (yashsd2@illinois.edu)

YTD - YouTube Descriptor

Functions and Users: We propose to implement a web based application (a browser extension) that can extract information from youtube videos and provide users with a short summary. Powered by keyword extraction and LLMs, our application will help users by providing short descriptions and relevant tags for their top-K youtube search results. This application is intended for all youtube users from students to the elderly, helping them save time and effort on finding the content of their choice.

Significance: In this day and age, if you want to watch a YouTube video for a particular topic there are so many options to choose from. This can be overwhelming and the number of views is not a very good indicator whether the content you need is covered in the video. Another issue with current YouTube search is that videos with a catchy thumbnail have a tendency to get higher views that creates a rolling stone effect. Our tool helps to mitigate this by providing a brief summary on the go. Some of the videos can be quite long and here our tool can provide a short summary so users can make an informed decision. Moreover, given that youtube feeds can be filled with videos about varying topics and misguided thumbnails, our key-word extraction feature can be useful to tag videos based on their content.

Approach: We will use open-source LLMs to process and summarize video content. Specifically, we will leverage APIs such as OpenAI's GPT (or any equivalent) for natural language understanding and summarization tasks. For keyword extraction, we will use NLP libraries such as NLTK or spaCy to identify and tag the main topics of each video. To interact with YouTube and fetch video details, we will utilize the YouTube Data API V3. Our backend will be developed using Flask, a lightweight Python web application framework, allowing for easy API endpoint creation and management. The JS-based frontend will interact with our backend services via REST calls, providing a seamless user experience without needing to leave the YouTube page.

We anticipate potential risks including API rate limits (for both YouTube and the LLMs) and changes to YouTube's page structure affecting how our extension injects content into the page. To mitigate these risks, we plan to implement caching mechanisms for API responses and regularly update our extension in response to changes in YouTube's frontend architecture.

Evaluation: To validate our tool's effectiveness and accuracy, we will employ a mixed-method evaluation. Through A/B testing, we'll quantitatively assess the efficiency gains in video search times with our extension. Qualitative insights will be gleaned from user feedback on summary accuracy and the enhancement of their browsing experience. We'll benchmark our algorithms' performance against manually summarized videos to establish precision, recall, and F1 scores.

Additionally, a user study will gauge the perceived quality of our summaries, inviting participants to score them against original content and manual benchmarks, ensuring a comprehensive and concise evaluation of our system's utility and precision.

Timeline:

- Week 1 (April 10th - April 16th): Planning and Initial Development
- Week 2 (April 17th - April 23rd): Backend and Algorithm Development
- Week 3 (April 24th - April 30th): Testing and Refinement
- Week 4 (May 1st - May 7th): Evaluation and Final Adjustments

Task division

- **Pranav Gor -** Focuses on backend development, including server setup, API integration and keyword extraction algorithms.
- **Yash Damania -** Responsible for frontend development, user interface design, and ensuring the extension is user-friendly and functional along with back-end for implementation of summarization.