IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 15, May 2023

Youtube Transcript Summarizer – Web Extension

Manal Wathore, Vedanti Chavan, Deep Gavate, Prof. Ninad More, Dr. Prakash Patil,
Department of Information Technology
D Y Patil College of Engineering, Pune, India

Abstract: *YouTube has become a crucial platform for news and information, and educational videos are highly popular. However, watching long YouTube videos can be time-consuming and tedious. Many videos employ attention-grabbing tactics like misleading thumbnails and advertisements, making it difficult to find relevant information. With the increasing number of YouTube users, there is a greater chance of encountering inaccurate or misleading content due to creators' desire for views. This can waste users' time and resources*

To address this issue, a project has been developed to provide quick, precise, and informative summaries of YouTube videos. The project utilizes a hugging face transformer and a Python API to extract the video's subtitles. The transformer model then performs text summarization on the subtitles to generate a summary. This enables users to save time by reading the summarized content instead of watching the entire video.

To enhance user interaction, a Chrome extension has been created. The extension features a "summarize" button, which, when clicked, displays the summarized text of the current YouTube video being played on the Google Chrome web browser. This user-friendly approach aims to facilitate the consumption of concise video summaries and improve the overall user experience.

Keywords: YouTube, hugging face transformer, Python API, Chrome extension, informative summaries

I. INTRODUCTION

In today's digital age, information can be accessed through various platforms such as social media, radio channels, and news channels. YouTube, in particular, has emerged as a major source of content consumption, catering to both educational and entertainment needs of users worldwide. However, users often encounter challenges while navigating through the vast amount of content available on the platform. Network issues often hamper the viewing experience, leading to low-quality videos with blurry visuals and frustrating loading times. Additionally, clickbait videos, created solely for the purpose of earning revenue, often lack substantial information, further exacerbating the problem. It becomes tedious and time-consuming to search for videos that contain the specific information users are seeking. Long videos with lengthy monologues make it difficult to find the relevant content unless the entire video is watched. produce a special electrical code, the RFID reader transmits an encoded radio signal. Python offers several packages that can simplify the process of accessing YouTube content, including transcripts of videos, through its APIs. Leveraging these capabilities, one can directly access the video's transcript and summarize it for the user. The Hugging Face transformer, a text summarization technique, proves to be a valuable tool for implementing this approach. To enhance user interaction and functionality, Chrome extensions can be employed. These extensions feature a "summarize" button, which, when clicked, displays the summarized text of the currently playing YouTube video on the Google Chrome web browser. The Hugging Face transformer package generates this summarized text. YouTube videos typically rely on manual descriptions and thumbnails for summarization, and given its immense popularity as the second most visited website globally, the proposed project aims to leverage the transformer package to provide meaningful and relevant video summaries.

Existing system

Automatic summarization is a valuable technique for condensing large amounts of information into concise summaries. In this research, the focus is on applying automated text summarization methods to YouTube videos by transcribing and summarizing their content.

The term frequency-inverse document frequency (TF-IDF) method is employed in this study to extract important keywords for the summary. This method considers the number of words and sentences in the text to identify significant

DOI: 10.48175/568

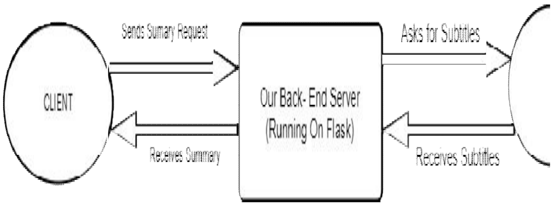
terms. Videos that are long or lack efficiency in delivering information often pose challenges for viewers who are looking for specific content within a limited time frame.

The proposed system aims to address this challenge by providing a summarized text that highlights essential information from lengthy videos. This allows students and researchers, who may have limited time, to extract useful data without watching the entire video. The evaluation of the system's results is performed using the Rouge method on the CNN-Daily Mail-Master dataset.

The TF-IDF method is based on determining the importance of a word in a given document. The value of TF-IDF increases with the frequency of the word's occurrence in the document. This approach utilizes weighted term frequency and inverse sentence frequency. Sentence vectors are scored based on their similarity to the query, and the highest scoring sentences are selected for the summary.

Summarization using TF-IDF is query-specific, assuming that sentences containing more specific words are relatively more important. Typically, target words are nouns. The method compares the term frequency (tf) in a document (where each sentence is treated as a document) with the document frequency (df), which represents the number of times a word occurs across all documents. The TF-IDF score is calculated accordingly.

II. PROPOSED SYSTEM



One common challenge faced by existing YouTube transcript summarizers is their limited ability to generate summaries when subtitles are in languages other than English. To address this issue, our project specifically focuses on providing support for various languages, allowing users to select the desired language for generating summaries. We have implemented a feature that enables users to obtain transcript files in multiple languages.

In addition to generating text-based summaries, our summarizer goes a step further by converting the summarized text into speech. This functionality aims to assist individuals with disabilities or the elderly who may have difficulties reading or hearing. By converting the summary to speech, we reduce the time required for users to consume the content fully. Users can use the summarizer to generate an audio file that condenses the video into a few minutes of spoken content derived from the summarized text.

The real-time applications of our project include:

- Transcribing the video from the provided link in an abstractive manner.
- Offering a simple and user-friendly interface for enhanced convenience.
- Minimizing user effort in obtaining the contents of a YouTube video without having to watch it in its entirety.

The web extensions block diagram provides a comprehensive overview of the interaction between web extensions and the different components of a web browser. At the core of this diagram is the browser itself, which serves as the platform for running web extensions. These extensions are small software programs designed to enhance the browser's functionality by introducing new features or modifying existing ones.

The key components depicted in the diagram include the Web Extension API, which acts as a bridge between the browser and the extensions. This API allows extensions to access and manipulate various browser features such as tabs, bookmarks, history, and user preferences. It facilitates seamless communication and interaction between extensions and the browser.

Another essential component is the Extension Manifest, which is a JSON file that provides vital information about the extension, including its name, version, required permissions, and utilized scripts or resources. The manifest serves as a blueprint for the browser to correctly load and understand the extension, ensuring its proper functioning.

Web Content Scripts form an integral part of web extensions, enabling the injection of scripts into web pages to modify their behavior or appearance. These scripts empower extensions to interact with web page content, manipulate elements in the DOM, and respond to user actions. They facilitate the customization and enhancement of web browsing experiences.

Background Scripts or event pages are scripts that run in the background of the browser, independent of any specific web page. They handle various tasks, including monitoring browser events, managing extension lifecycle events, and overseeing long-running processes or background activities. These scripts ensure the seamless functioning of extensions behind the scenes.

User Interface (UI) components are responsible for presenting the extension's user interface elements to users. This includes browser action buttons, options pages, and pop-up windows that enable users to interact with the extension and access its features conveniently. The UI components enhance the usability and accessibility of web extensions.

Additional components highlighted in the block diagram include Extension Storage, which provides storage capabilities for extensions to store and retrieve data, and the Content Security Policy (CSP), which safeguards against security vulnerabilities and ensures that extensions comply with established security policies. These components contribute to the overall functionality and security of web extensions.

In summary, the web extensions block diagram provides a meaningful representation of the interplay between web extensions and various components of a web browser. It showcases the vital roles these components play in extending the capabilities of web browsers and enhancing the browsing experience for users.

III. RESULT

The YouTube Transcript summarizer developed in this project provides an efficient solution for users to save time and resources. By utilizing the Chrome extension of the Google Chrome browser, users can simply click on the summarize button to access the transcripts of the YouTube video. This is achieved through the integration of a Python API that retrieves the necessary transcripts. The retrieved transcripts are then summarized using the transformers package, which generates a concise summary. The summarized text is displayed to the user within the chrome extension web page, allowing them to quickly grasp the main points of the video without the need to watch it in its entirety.

This project offers several benefits to users. Firstly, it significantly saves time by eliminating the need to watch lengthy videos. Users can obtain the essence of the content within seconds, enabling them to make informed decisions about whether or not to invest more time in watching the complete video. Additionally, the summarizer serves as a useful tool for identifying potentially inappropriate or harmful content. By providing a summary, users can assess the nature of the video without being exposed to content that may disturb their viewing experience.

Furthermore, the project ensures a seamless user interface experience by leveraging the capabilities of chrome extensions. Users can access the summarized text without the inconvenience of copying and pasting URLs or relying on third-party applications. This user-friendly approach enhances accessibility and streamlines the process of obtaining video summaries.

Overall, the YouTube Transcript summarizer project offers an effective and convenient solution for users to quickly extract the essential information from YouTube videos, ultimately improving their browsing experience and enabling them to make more informed decisions about their content consumption.

Conclusion

YouTube's immense user base and the ever-growing volume of video content being uploaded necessitate more efficient methods of information retrieval. This paper proposes a novel approach to video summarization by utilizing the subtitles of YouTube videos. By focusing on text-based summarization techniques, the process becomes accessible to a wider range of users, eliminating the need for advanced technical knowledge and high processing power. The research explores extractive and abstractive methods of text summarization, highlighting various models used in abstractive summarization. The ultimate goal of this project is to develop user-friendly software that automates the generation of video summaries, enabling users to grasp the essence of the content before committing to watching the entire video. This innovation empowers YouTube users to efficiently extract insights and make informed decisions without the burden of extensive viewing, enhancing their overall experience on the platform.

ACKNOWLEDGMENT

We express our sense of gratitude towards our project guide Prof. Ninad More and our project co-ordinator Dr. Prakash Patil for the valuable guidance at step of study of this project and also for the necessary guidelines and timely co-operation during the completion of project.

REFERENCES

[1] Begum1, N. Musrat Sultana2, Dharma Ashrith, “YouTube Transcript Summarizer” Volume 10.

[2] S. Tharun, R. Kranthi Kumar, P. Sai Sravanth, G. Srujan Reddy, B. Akshay, “Survey on Abstractive Transcript Summarization of YouTube Videos”, Volume 2.

[3] Shraddha Yadav, Arun Kumar Behra, Chandra Shekhar Sahu, Nilmani Chandrakar, “Summary and Keyword Extraction from Youtube Video Transcript”, Volume 3 [4]https://huggingface.co/transformers/installation.html https://atmamani.github.io/blog/buildingrestful-apis-with-flask-in-python/

[5]https://pypi.org/project/youtube-transcript-api/ https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask

[6]https://medium.com/swlh/parsing-rest-api-payload-and-query-parameters-with-flask-betterthan-marshmallow-aa79c889e3ca [7]https://betterprogramming.pub/the-ultimate-guide-to-building-a-chrome-extension4c01834c63ec

[8] https://developer.chrome.com/docs/extensions/mv2/