

Youtube Transcript Summarizer Using Flask And Nlp

P. Vijaya Kumari^{1*}, M. Chenna Keshava², C. Narendra³, P. Akanksha⁴, K. Sravani⁵

**1, 2 Assistant Professor (A), Department of Computer Science and Engineering, Jawaharlal Nehru Technological University College of Engineering, Pulivendula, Andhra Pradesh, India, E-mail:- vijaya.jntuacep@gmail.com¹, keshava1047@gmail.com²*

3, 4, 5 Student (U.G), Department of Computer Science and Engineering, Jawaharlal Nehru Technological University College of Engineering, Pulivendula, Andhra Pradesh, India, E-mail:- cnarendra329@gmail.com³, akanksha17p@gmail.com⁴, sravani0602@gmail.com⁵

Abstract

The project's goal is to design a user interface that will improve user experience, allow flexibility for downloading transcript summary files, and automate WhatsApp and email. Throughout the day, an enormous number of video recordings are generated and shared on the Internet. It has become quite difficult to devote time to watching movies that may last longer than expected, and our efforts may be in vain if we are unable to extract useful information from them. Summarizing transcripts of such movies automatically allows us to rapidly spot essential patterns in the video, saving us time and effort from having to go through the entire content. We use Python APIs for text transcription in this project. The transcript is then summarized using natural language processing (NLP). User Interface: HTML, CSS, JS, and Bootstrap For our User Interface, we use Flask for Python as the backend. The user can download the summarized forms, such as pdf and word, and share his summarized transcript by email and WhatsApp. This project will provide us with practical experience using NLP approaches for abstractive text summarizing.

Keywords: - Summarization, Natural Language Processing, CNN, Hugging-Face Transformers.

I. INTRODUCTION

Using the Flask framework, this backend takes API calls from the client and answers with a summary text response. This API can only be used with YouTube videos that have closed captions that have been properly prepared. The Summarizer is also available online, where users may make basic API calls and read the results on a webpage.

This backend accepts API calls from the client and responds with a summary text response using the Flask framework. This API can only be used with YouTube videos that have been correctly prepared closed captions. Users can also utilize the Summarizer online, where they can execute basic API calls and view the results on a webpage.

YouTube

Is a free online video sharing and social media platform that launched in the United States in February 2005. It is now one of the most popular video platforms, with subscribers watching over 1 billion hours of video per day. Closed captions are text retrieved from a video that is intended to give the audience with more information (such as conversations, speech translation, and non-speech features). They're widely used to understand video

without having to listen to it in order to save time.

There are several videos with transcripts on YouTube. Summarization is especially effective when films are longer and different segments have

Varying degrees of importance. In this case, video summary may be advantageous in terms of saving time for the viewer. It will boost user productivity by allowing them to concentrate entirely on the video's relevant text. It gets its information from video subtitles.

We'd be developing a capacity for summarizing YouTube movies with captions given by the owner, and generating a summarized text answer utilizing different summary methods. We will generate summaries in a variety of forms because the Summarizer should be straightforward to use for the end user.

We'll set up a Flask back-end server to receive GET requests with the YouTube Video Id, Algorithm Choice, and Response Ratio. This server will prevent the user from performing any summary processing.

This approach also has the capability of immediately improving the algorithm in the future;

users do not need to update on their end, saving time and money. To obtain the summary, the client may gain access to the API.

NLP is an area of artificial intelligence that explores how humans and machines communicate. Making video transcript summaries is a time-consuming operation. Longer videos can be described in a concise, fluent, and, most importantly, correct manner.

The main idea is to be able to choose a small subset of the most significant facts from the entire collection and present it in a simple manner.

A user-friendly layout As the amount of textual material on the internet expands, automatic text summarising techniques are becoming increasingly common. It has the potential to be tremendously beneficial because more useful information may be read in less time. Putting together summaries.

Because machines must understand what humans have written and produce human-readable results, NLP is concerned with transcripts.

II. LITERATURE SURVEY

From [1], author proposed two different methods to generate summary and important keywords from the given YouTube video - extractive and abstractive. They have made a simple user interface through which users can easily get their summaries through these methods, and surely find it easy to interact with their user interface and get what they want. Their project surely satisfies the users and solve all the problems that it's supposed to tackle which is saving time and efforts, by providing only the useful information about the topic which interests them so that they don't have to watch those long videos and the time that saved can be used in gaining more knowledge.

In [2], authors propose a video summarizing system based on natural language processing (NLP) and Machine Learning to summarize the YouTube video transcripts without losing the key elements. The quantity of videos available on web platforms is steadily expanding. The content is made available globally, primarily for educational purposes. Additionally, educational content is available on YouTube, Facebook, Google, and Instagram. A significant issue of extracting information from videos is that unlike an image, where data can be collected from a single frame, a viewer must watch the entire video to grasp the context. The suggested method involves retrieving

transcripts from the video link provided by the user and then summarizing the text by using Hugging Face Transformers and Pipelining. The built model accepts video links and the required summary duration as input from the user and generates a summarized transcript as output.

The proposed method in [3] is focuses on the recent advances in the area and provides a comprehensive survey of the existing deep-learning-based methods for generic video summarization. After presenting the motivation behind the development of technologies for video summarization, they formulated the video summarization task and discuss the main characteristics of a typical deep-learning-based analysis pipeline. Then, suggested a taxonomy of the existing algorithms and provide a systematic review of the relevant literature that shows the evolution of the deep-learning-based video summarization technologies and leads to suggestions for future developments.

According to [4], previous methods mainly take diversity and representativeness of generated summaries as prior knowledge in algorithm design. In this paper [4], they formulate video summarization as a content-based recommender problem, which should distill the most useful content from a long video for users who suffer from information overload. A scalable deep neural network is proposed on predicting if one video segment is a useful segment for users by explicitly modelling both segment and video. Moreover, they accomplished scene and action recognition in untrimmed videos to find more correlations among different aspects of video understanding tasks. Also, paper discussed the effect of audio and visual features in summarization task.

From [5], we can conclude that, video summarization and skimming has become an indispensable tool of any practical video content management system. This paper [5] provides a tutorial on the existing abstraction work for generic videos and presents state-of-the-art techniques for feature film skimming. The paper also describes the authors' recent work on movie skimming using audiovisual tempo analysis and specific cinematic rules. With the maturity of the movie genre classification, content understanding and video abstraction techniques, an automatic movie content analysis system that facilitates navigation, browsing, and search of desired movie content is possible soon.

As per [6], Automatic summarization techniques

will give the user an easy way to look up important content of a collection of media and to browse media of their choice later. With the evolution of sophisticated capturing devices, cloud-based summarization solutions, which have a lot of turnaround time, are less preferred by end user. In this paper, author proposed a real-time video summarization technique for mobile platform which analyses the video during live camera recording and generates summary instantaneously. This technique employs the method of analyzing intrinsic video data like the contents of video stream, and corresponding extrinsic metadata such as external camera information of the video stream. The proposed technique has been able to achieve an f-measure of 0.66 and 0.84 on SumMe and SumLive datasets respectively while limiting the overall power consumption to 20 milliamps on an embedded system.

In [7], Authors propose online video highlighting, a principled way of generating short video summarizing the most important and interesting contents of an unedited and unstructured video, costly both timewise and financially for manual processing. Specifically, their method learns a dictionary from given video using group sparse coding, and updates atoms in the dictionary on-the-fly. A summary video is then generated by combining segments that cannot be sparsely reconstructed using the learned dictionary. The online fashion of their proposed method enables it to process arbitrarily long videos and start generating summaries before seeing the end of the video. Moreover, the processing time required by proposed method is close to the original video length, achieving quasi real-time summarization speed.

From [8], a generic framework of a user attention model, which estimates the attentions viewers may pay to video contents is proposed. As human attention is an effective and efficient mechanism for information prioritizing and filtering, user attention model provides an effective approach to video indexing based on importance ranking. In particular, they defined viewer attention through multiple sensory perceptions, i.e. visual and aural stimulus as well as partly semantic understanding. Also, a set of modeling methods for visual and aural attentions are proposed. As one of important applications of user attention model, a feasible solution of video summarization, without fully semantic understanding of video content as well as complex heuristic rules, is implemented to demonstrate the effectiveness, robustness, and generality of the user attention model. The

promising results from the user study on video summarization indicate that the user attention model is an alternative way to video understanding.

According to [9], argued that video summarization would benefit from greater incorporation of external information, particularly user-based information that is unobtrusively sourced, in order to overcome longstanding challenges such as the semantic gap and providing video summaries that have greater relevance to individual users.

In [10], In this paper, authors surveyed the video classification literature. They find that features are drawn from three modalities—text, audio, and visual—and that a large variety of combinations of features and classification have been explored. They also described the general features chosen and summarize the research in this area. We conclude with ideas for further research.

III. PROBLEM STATEMENT

Throughout the day, an increasing number of video recordings are generated and shared on the Internet. It has become quite difficult to devote time to watching movies that may last longer than expected, and our efforts may be in vain if we are unable to extract useful information from them. Summarizing transcripts of such movies automatically allows us to rapidly spot essential patterns in the video, saving us time and effort from having to go through the entire content.

IV. WORK FLOW

The steps are followed to accomplish the transcript information:

1. Client sends request to our flask backend server
2. Flask backend server asks for subtitles from the YouTube using YouTube API.
3. YouTube sends the subtitles to our server and text summarization is done in our backend server
4. Client receives the summarized text

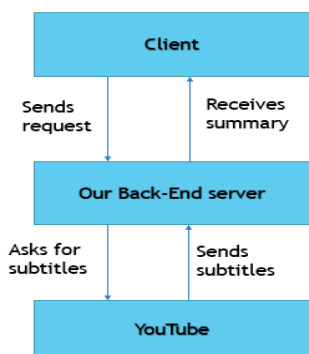


Fig. 1: Workflow of our project

Here the user gives the YouTube link for transcription. Based on the YouTube URL the subtitles are fetched by the backend flask server from YouTube server. The subtitles that are extracted from the YouTube are given to a default CNN model in our flask backend server. And the summarized text will be received by the client. In our application the user is allowed to download the summarized text in different file formats and can also translate the transcript text into different languages provided this was done using google translation system.

V. RESULTS ANALYSIS

Summarized transcript generation using NLP and Flask :

In Fig. 2, the homepage of our project YouTube transcript summarizer with navigation bar of three links home, transcript now, contact us which will redirect to respective pages

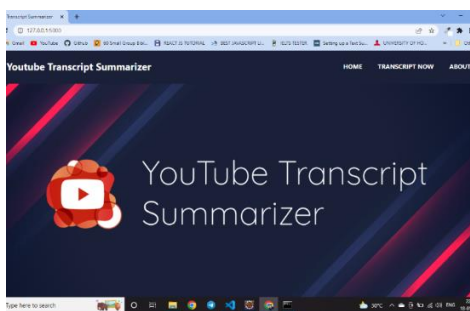


Fig. 2: Homepage of YouTube transcript summarizer

In Fig. 3 the user interface is shown for the user with the form which accepts the YouTube link. The transcript now button will be there on clicking that the link will be send to backend using flask.

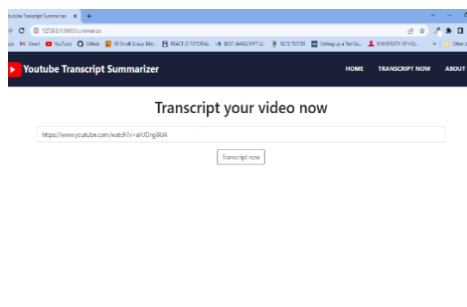


Fig. 3: User-Interface for accepting the YouTube video link

In Fig. 4 After clicking on transcript now button the summarized text is generated at backend and displayed in the summarized transcript page.

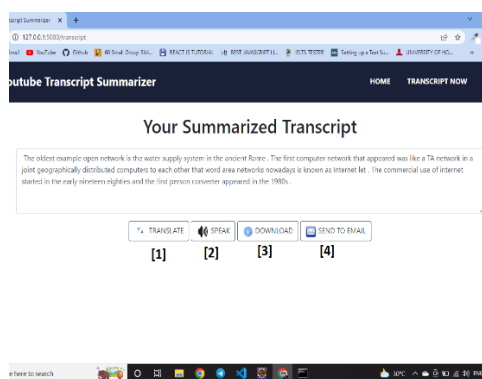


Fig. 4: Displaying the summarized text

After summarization of text, the user is provided with four options (buttons).

- [1] **Translate:** user can translate their summarized text into different languages provided. Fig 5 and Fig 6 shows the output for translating the summarized text into telugu language
- [2] **Speak:** if user is unable to read the transcript can use this option, clicking on speak button will speak aloud the summarized transcript of any language.
- [3] **Download:** user can download their summarized text into different file formats. Fig 7 shows the output for downloading the summarized text into different file format
- [4] **Send to mail:** If user want to share the transcript file to his/her mail or want to share with their friends can use this. Fig 8 shows the output for sending the summarized text file.

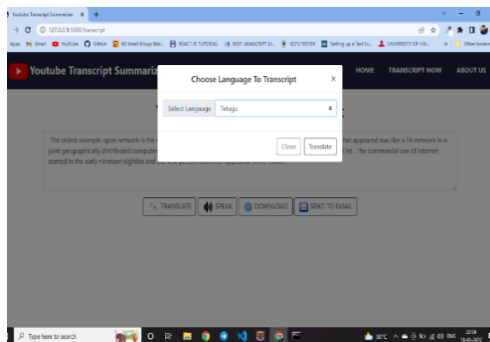


Fig. 5: UI for selecting the desired language to transcript

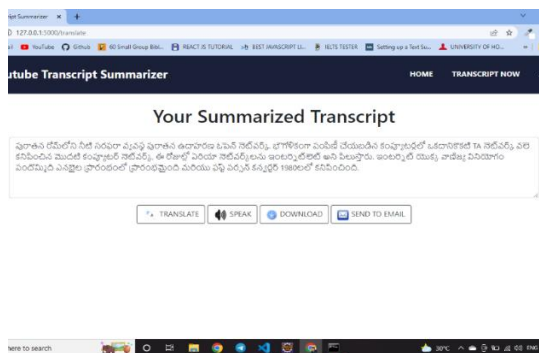


Fig. 6 Summarized text translated to desired language

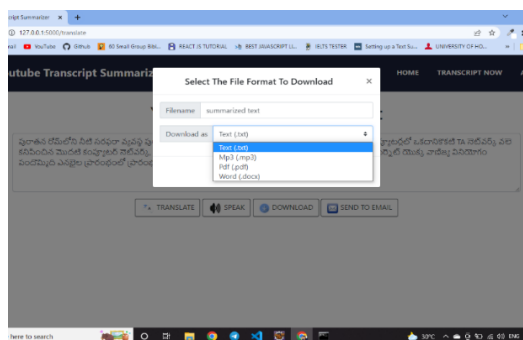


Fig. 7 Prompt allowing user for downloading file in different formats

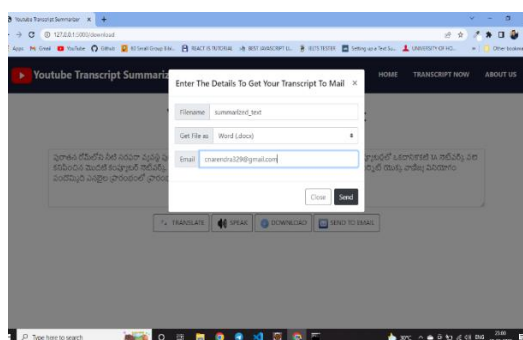


Fig. 8 Prompt allowing user for sending file to email in different formats

VI. CONCLUSION

We developed a system for transcribing YouTube videos, as well as a platform that summaries the transcript. We created a system with a simple user

interface and a lot of features. We have made it possible for users to obtain their transcript files in many languages. Additionally, users can obtain a transcript file in a variety of formats. We created this system for folks who have trouble reading by including alternatives to speak and download as mp3 files. Using the send mail option, the user can send the transcript file to his or her own or any other email address. In total, we created a summarizing transcribing system with a user interface and numerous features.

Real Time Applications

- ✓ Transcripts the video from the given link abstractively.
- ✓ Allows the user to translate the transcript file in different languages provided.
- ✓ Allows user to download the transcript file in different file formats.
- ✓ Provided the simple user interface for user convenience.
- ✓ Decreases the efforts of user to know the contents of the YouTube video without watching

Limitations

- Transcript cannot get from the videos without subtitle
- Translated text other than English wont support text and pdf file formats because of encoding format

REFERENCES

- [1]. Shraddha Yadav, Arun Kumar Behra , Chandra Shekhar Sahu, Nilmani Chandrakar, “ SUMMARY AND KEYWORD EXTRACTION FROM YOUTUBE VIDEO TRANSCRIPT”, International Research Journal of Modernization in Engineering Technology and Science Volume:03/Issue:06/June-2021 Impact Factor- 5.354 .
- [2]. A. N. S. S. Vybhavi, L. V. Saroja, J. Duvvuru and J. Bayana, "Video Transcript Summarizer," 2022 International Mobile and Embedded Technology Conference (MECON), 2022, pp. 461-465, doi: 10.1109/MECON53876.2022.9751991.
- [3]. E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris and I. Patras, "Video Summarization Using Deep Neural Networks: A Survey," in Proceedings of the IEEE, vol. 109, no. 11, pp. 1838-1863, Nov. 2021, doi:10.1109/JPROC.2021.3117472.

- [4]. Yudong Jiang, Kaixu Cui, Bo Peng, Changliang Xu; "Comprehensive Video Understanding: Video Summarization with Content-Based Video Recommender Design"; Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 0-0.
- [5]. Ying Li, Shih-Hung Lee, Chia-Hung Yeh and C. J. Kuo, "Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques," in IEEE Signal Processing Magazine, vol. 23, no. 2, pp. 79-89, March 2006, doi: 10.1109/MSP.2006.1621451.
- [6]. P. Choudhary, S. P. Munukutla, K. S. Rajesh and A. S. Shukla, "Real time video summarization on mobile platform," 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017, pp. 1045-1050, doi: 10.1109/ICME.2017.8019530.
- [7]. Bin Zhao, Eric P. Xing; Quasi Real-Time Summarization for Consumer Videos; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2513-2520.
- [8]. Yu-Fei Ma, Xian-Sheng Hua, Lie Lu and Hong-Jiang Zhang, "A generic framework of user attention model and its application in video summarization," in IEEE Transactions on Multimedia, vol. 7, no. 5, pp. 907-919, Oct. 2005, doi: 10.1109/TMM.2005.854410.
- [9]. Video summarization: A conceptual framework and survey of the state of the art, Journal of Visual Communication and Image Representation, Volume 19, Issue 2, 2008, Pages 121-143, Arthur G. Money, Harry Agios, -143, ISSN 1047-3203.
- [10]. D. Brezeale and D. J. Cook, "Automatic Video Classification: A Survey of the Literature," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 3, pp. 416-430, May 2008, doi: 10.1109/TSMCC.2008.91