

YouTube Transcript Summarizer using Natural Language Processing

Prof. S. H. Chaflekar¹, Achal Bahadure², Hosanna Bramhapurikar³, Ruchika Satpute⁴,
Rutuja Jumde⁵, Sakshi A. Bakhare⁶, Shivani Bhirange⁷

Assistant Professor, Department of Information Technology¹

Students, Department of Information Technology^{2,3,4,5,6,7}

Priyadarshini Bhagwati College of Engineering, Nagpur, Maharashtra, India

snehalchaflekar@gmail.com¹, achalbahadure001@gmail.com², bramhapurikarhosanna@gmail.com³,
ruchikasatpute345@gmail.com⁴, rutujajumde5@gmail.com⁵, sakshibakhare@gmail.com⁶, shivanibhirange@gmail.com⁷

Abstract: *We spend a noticeable amount of our weekly time watching YouTube videos, be it for entertainment, education, or exploring our interests. In most cases, the overall intent is to obtain some form of information from the video. We were seeking a solution to increase the efficiency of this "information extraction" process as YouTube's speed adjustment option is the only relevant tool. The summarizer is a Chrome extension that works with YouTube to extract the key points of a video and make them accessible to the user. The summary is customizable per user's request, allowing varying extents of summarization. Key points from the summarization process, together with corresponding time-stamps, are then presented to the user through a small UI next to the video feed. This allows the user to navigate to more important sections of the video, to get to the key points more efficiently. The main idea behind it is to be able to find a short subset of the most essential information from the entire set and present it in a human-readable format. As online Textual data grows, automatic Summarization of text methods has the potential to become very helpful because more useful information can be read in a short time.*

Keywords: Natural Language Processing, Automatic text summarization, Extractive approach, Abstractive approach, Speech Summarization.

I. INTRODUCTION

In this paper, we are going to make a summarized version of a YouTube transcript. Enormous number of video recordings are being created and shared on the Internet throughout the day. It has become really difficult to spend time in watching such videos which may have a longer duration than expected and sometimes our efforts may become futile if we couldn't find relevant information out of it. Summarizing transcripts of such videos automatically allows us to quickly look out for the important patterns in the video and helps us to save time and efforts to go through the whole content of the video.

In this milestone, we are going to utilize a python API which allows you to get the transcripts/subtitles for a given YouTube Video. It also works for automatically generated subtitles, supports translating subtitles and it does not require a headless browser, like other selenium-based solutions do!

This project is an integration of web development and the very emerging technology, machine learning. This project aims to provide summarized documentation of a video that are too long to be watched. Today, education is more dependent on online sources rather than the offline source, and no one has much time to spent on lecture videos that are too long to watch. So, to resolve this, there should be a tool which can provide a summarization of the video and therefore save time.

Natural Language Processing (NLP) is a field of Artificial Intelligence that focuses mainly on the study of the interaction between human languages and machines. Generating summaries of video transcripts is the process of generating short, fluent, and most importantly accurate summaries of longer videos.

II. LITERATURE SURVEY

1. “Automated video summarization using speech transcript”, Cuneyt M. Taskiran, Arnon Amir, Dulce B. Ponceleon, Edward J. Delp.

This research describes the compact representations of video data can enable efficient video browsing. Such representations provide the user with information about the content of the particular sequence being examined while preserving the essential message. They propose a method to automatically generate video summaries for long videos.

2. “Video Summarization using NLP”, Sanjana R, Sai Gagana V, Vedhavathi K R, Kiran K N.

This paper proposes an automatic video summarization algorithm using NLP based algorithms. With an increase in internet videos on the video repository platforms like YouTube, Instagram etc. there is an increase in demand for good summarization algorithms to summarize various videos. This paper aims to produce short and concise video summary that summarizes various YouTube videos. The proposed technique first summarizes the YouTube video transcripts based on which summarized video is generated.

3. “Summary and Keyword Extraction from Youtube video Transcript”, Shraddha Yadav, Arun Kumar Behra, Chandra Shekhar Sahu, Nilmani Chandrakar.

This research paper aims at extracting the summaries from video transcripts and also generating important keywords from it which will use Natural Language Processing methods for extractive and abstractive summarization. So, generating the summaries of those video transcripts will save you lots of time and you will quickly gather more useful and important information from it, which will surely save your efforts and time to watch the whole video.

4. “Digital video Summarization Techniques”, Ashenafi Workie, Rajesh Sharma, Yun Koo Chung.

This paper aims to evaluate a system that automatically summarizes video files (image and audio), it should be taken into account how the system works and which are the part of the process that should be evaluated, as two main topics to be evaluated can be differentiated: the video summary and the text summary. So, in the present article it is presented a complete way in order to evaluate this type of systems efficiently. With this objective, the authors have performed two types of evaluation: objective and subjective (the main focus of this paper).

5. “VSCAN: An Enhanced Video Summarization using Density-based Spatial Clustering”

In this paper, a modified version of an evaluation method Comparison of User Summaries (CUS) is used to evaluate the quality of video summaries. The modifications proposed to CUS method aims at providing a more perceptual assessment of the quality of the automatic video summaries.

6. “A Pertinent Evaluation of Automatic Video Summary”

They propose an effective method for identifying the true matches between AT (Automatic Summary) and GT (Ground Truth User Summary) for the performance evaluation of the summarised videos. It includes the initial establishment of matched frames via two-way search followed by a consistency check where weak and false matches are eliminated.

III. PROPOSED METHODOLOGY

Proposed System

3.1 NLP and Hugging Face Algorithm

Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyse large amounts of natural language data. The increase in popularity of video content on the internet requires an efficient way of representing or managing the video. This can be done by representing the videos on the basis of their summary. Hugging Face provides models for a variety of tasks. Some of them are: In NLP: Summarization, question answering, table question answering, text classification, fill mask. Sentence similarity, translation, token classification, feature extraction.

The app could detect emotions and answer questions based on the context and emotions. Hugging Face aims to become GitHub for machine learning. Hugging Face is one of the leading startups in the NLP space.

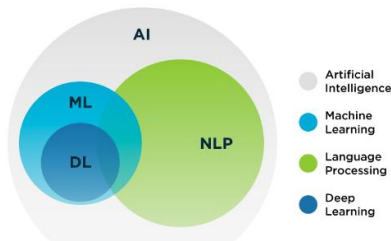


Figure 1: NLP and Algorithm Integration

3.2. Modules

Our system development has been mainly done in 3 modules that are as follows:-

1. Getting Started With Speech To Text Transcript
2. Perform Text Summarization
3. Build a User Interface for Extension Popup

The modules of our project are as follows:

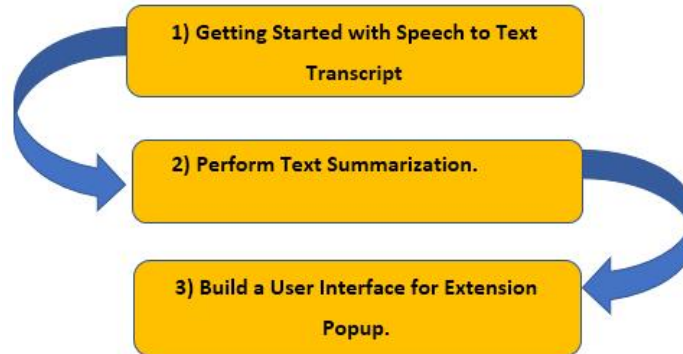


Figure 2: Project Modules

MODULE 1: Getting Started With Speech to Text Transcript

First up all we have import Some Python Basic required libraries like, natsort, tkinter, from tkinter import file dialog, speech_recognition, numpy from moviepy.video.io. Video File Clip import Video File Clip, pydub.

In This Module, We have Simply convert audio to text form by using Python libraries on the basis of NLP Techniques. Natsort is the natsorted identifies numbers anywhere in a string and sorts them naturally. For numerical data we have used numpy. numpy is a Python library used for working with arrays. For manipulate audio with a simple and easy high level interface we have used Pydub. Python provides a module called pydub to work with audio files. pydub is a Python Library to work with only .wav files. A function that splits the audio file into chunks. For performing voice recognition, with support for several engines and APIs we have used speech_recognition library.

Here, we got to know, How the NLP Technique works with Speech to text Transcript. Natural Language Processing (NLP) include speech to text is a profound application of Deep Learning which allows the machines to understand human language and read it with motive to act and react, as usual, humans do.

MODULE 2: Perform Text Summarization

Text summarization is the task of shortening long pieces of text into a concise summary that preserves key information content and overall meaning. There are two different approaches that are widely used for text summarization:

- Extractive Summarization: This is where the model identifies the important sentences and phrases from the original text and only outputs those.
- Abstractive Summarization: The model produces a completely different text that is shorter than the original, it generates new sentences in a new form, just like humans do. In this project, we will use transformers for this

approach. In this milestone, we will use Hugging Face's transformers library in Python to perform abstractive text summarization on the transcript obtained from previous milestone.

1. Applying Natural Language Processing on the Subtitles:

We have a video with subtitles. We applied an Automatic NLP based LSA summarization algorithm on the subtitle to generate the summary. Basically, we converted the subtitles of the video into a text document and then applied the summarization algorithm. Python library sumy provides the summary for a text document to the number of sentences you specify as argument. There are many summarization algorithms that we can use with the help of this library. But we have used the LSA algorithm.

2. Fitting the Duration which user Provides:

Using the python library sumy, it is possible to rank each sentence (or subtitles in our case). Each subtitle has a certain duration in the video. In order to fit the user duration, we found the average duration of each subtitle by dividing the Total duration of the video with the Number of subtitles.

Using this average duration, we have found the approximate number of sentences which we need to produce the summarized video. This summarization technique works in such a way that the top most ranked subtitles are taken into consideration for the final summarized video. If the total duration of the summarized subtitles is more, then it is possible to reduce the one that is least ranked and vice versa. In this way, it is possible to fit the video to the time provided by user.

3. Creating the Final Summarized Video:

So now we got the summary of the subtitles and now we have to generate the summarized video. We have used the python module called Moviepy. Using the time stamps in the summarized subtitles we divided the video into several segments and finally merged to create the final summarized video.

Hence, by following the above steps we were able to generate the video summarization for the given video. And these steps which we are implemented in Jupyter Notebook: -

First, we will be using Jupyter Notebook for NLP Techniques, we are going to call for some libraries like transformers, and then we will be assigning a variable to call for a YouTube vide via library of python YouTube API. We will be using the unique YouTube ID first to perform and get the whole audio to text print in out Jupyter notebook, after we have printed all the text, we will perform text summarisations

We will be using some methods to know the string which we printed and after that we will perform text summarisation to get the summary of any YouTube video.

After Generating the Summary of any Video, we will be making a separate .txt file where we are going to print all the summarised text in that file. Finally, the summarised text will be displayed both on Jupyter Notebook and the .txt file which we created.

MODULE 3: Build a User Interface for Extension Popup

Here, we have used some Python Libraries on the basis of NLP Technique. We have used this technique extensively in our project. Some NLP libraries like nltk, spacy, pandas, webvtt -py , youtube _dl , scikit – learn and many more.

In our Project we have created a GUI and for GUI building we have used python's basic library called Tkinter. Tkinter is the standard GUI Library for Python. Python when comined with Tkinter provides a fast and easy way to create GUI applications. For numerical data we have used numpy. numpy is a Python library used for working with arrays. & For importing youtube videos reading and getting transcript we have used Youtube -dl library. For the purpose of reading/writing WebTT caption files we have used webvtt-py python module.

First we will include our transcript YouTube video , we will provide URL of the video, after that we can specify the type of summariser which we are using like tlf based, frequency based, gensim based, and after that we will select our fraction according to our code which is 0.3 and after that we will provide a path where all the data will be stored, if all the information is correct then we will click on submit button, after clicking on submit button another open folder button will be shown, after clicking on the open folder button we can see our original text (corpus) and summarised text both in .txt extension.

The Process of Module 3 is as follows: -

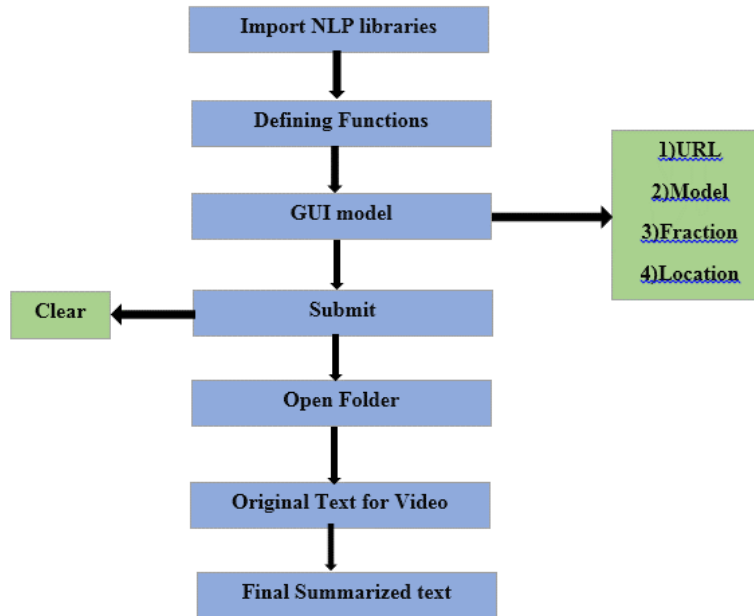


Figure 3: Module 3 Representation

IV. RESULT AND DISCUSSION

This project is used for following functions:

- Conferences – a fast outline can be generated from your team conferences and video conferences, to be employed in more conferences or additional intricate discussions.
- Patents research - how to extract the foremost vital claims across patents/research papers.
- Crash-Course – Students that watch YouTube videos to find out a theme topic quickly and concisely will get a quick read of the video and check if a connected course programme is present or not.
- Online Classes – Students that have missed classes during this online era of education will build notes from the summary of the video lecture. Also, notes can be simply obtained to be distributed to all the students.

The result of the project are as follows:

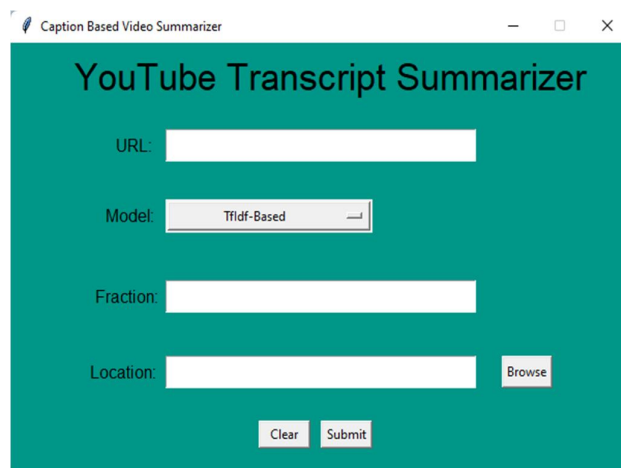


Figure 4: Get Input from the User.

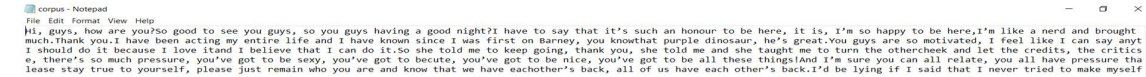


Figure 4.1: Showing the Original text for the given video.



Figure 4.2: Showing the Summarized text for the given video.

V. CONCLUSION

In this paper, we propose two different methods to generate summary and important keywords from the given YouTube video-extractive and abstractive. We have made a simple interface through which users can easily get their summaries through this method, and surely find it easy to interact with our user interface and get what they want. We are sure that our project will surely satisfy the user's 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 is saved can be used in gaining more knowledge.

REFERENCES

- [1]. Anuj Gupta, Bodhisattwa Prasad Majumder, Harshit Surana and Sowmya Vajjala. "Practical Natural Language Processing" A comprehensive guide to build Real World NLP System. 1st edition. O'Reilly Media, Inc. (June, 2020)
- [2]. Ankit Kumar, Zixin Luo & Ming Xu, "Text Summarization using Natural Language Processing" WPL Juniper Networks. (March, 2018).
- [3]. Korra Sathya Babu, Santosh Kumar Bharti and Sanjay Kumar Jena 2017. "Automatic Keyword Extraction for Text Summarization: A Survey". (Feb, 2017)
- [4]. Pratik K Biswas, Aleksander Lakubovich. (2020) "Extractive Summarization on Call Transcript" AI & Data Science, Global Network and Technology, Verizon Communication, New Jersey, USA.
- [5]. Cuneyt M. Taskiran, Armon Amir, Dulce Ponceleon and Edward J. Delp. 2009. "Automatic Video Summarization Using Speech Transcripts". Video and Image Processing Laboratory (VIPER).
- [6]. Hamza Shabbir Moiyadi, Harsh Desai, Dhairyaa Pawar, Geet Agrawal, and Nilesh M. Patil. (2016). "NLP Based Text Summarization Using Semantic Analysis".
- [7]. Feifan Liu, Deana Penneli, Fei Liu and Yang Liu. 2009. "Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts". Computer Science Department the University of Texas at Dallas Richardson, USA. (June 2009).
- [8]. Shraddha Yadav, Arun Kumar Behra, Chandra Shekhar Sahu, Nilmani Chandrakar. "Summary and Keyword Extraction from Youtube video Transcript". Department of Computer Science and Engineering, Government Engineering College, Raipur, Chhattisgarh, India. (June 2021).
- [9]. Gaurav Sharma, Shaba Parveen Khan, Shivanshu Sharma and Syed Ubed Ali (2021). "Summarizer For Easy Video Assessment (SEVA)". Vol.03. e-ISSN: 2582-5208. (April, 2021)
- [10]. Shengli Song, Haitao Huang and Tongxiang Ruan. 2018. "Abstractive Text summarization using LSTM-CNN based deep learning. Springer Science+Business Media. (Feb, 2018).
- [11]. Rada Mihalcea, Paul Tarau. "TextRank: Bringing Order into Texts". Department of Computer Science University of North Texas.
- [12]. Gang Liu, Jiabao Guo. 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. School of Computer Science, Hubei University of Technology, Wuhan, China.