ATHARVA EDUCATIONAL TRUST'S
ATHARVA COLLEGE OF ENGINEERING
(Approved by AICTE, Recognized by Government of Maharashtra)
Department of Computer Engineering
Academic Year 2021-22

"Youtube Transcript Summary Generation"

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By
Shriya Bijam Roll No: 10
Bharat Choudhary Roll No: 16
Nikita Dubey Roll No: 64

Supervisor
Prof. Alka Shrivastav

ATHARVA EDUCATIONAL TRUST'S
ATHARVA COLLEGE OF ENGINEERING
(Approved by AICTE, Recognized by Government of Maharashtra)
Department of Computer Engineering
Academic Year 2021-22

CERTIFICATE

This is to certify that the Mini Project entitled **"Youtube Transcript Summary Generation"** is a bonafide work of **Shriya Bijam (roll no-10), Bharat Choudhary (roll no-16), Nikita Dubey (roll no-64) ,** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering" .**

Prof. Alka Shrivastav
Supervisor

Dr. SUVARNA PANSAMBAL
Head of Department

Dr. Shrikant Kallurkar
Principal

Mini Project Approval

This Mini Project entitled **"Youtube Transcript Summary Generation"** by **Shriya Bijam (roll no-10), Bharat Choudhary (roll no-16), Nikita Dubey (roll no-64)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Place:

Contents

ABSTRACT:

With the rise of Internet reach worldwide, there has been a substantial increase in the number of videos uploaded and shared on various networking sites. One of the most popular video sharing sites is YouTube. On YouTube, numerous videos are uploaded every single day from all around the world. These videos include educational content, speeches, tutorials, so on. Compact representations of video data can enable efficient video browsing. Such representations provide the user with information about the content of the particular video being examined while preserving the essential message. Hence, generating the summary of a video prior to watching it, will tell you the gist of the video, on the basis of which you can decide if it is worth watching the entire video. This demo proposes YO-script, a Chrome extension, which extracts the summaries from video transcripts and generates important keywords from it. The project uses Natural Language Processing methods for extractive and abstractive summarization.

Keywords: Natural Language Processing, Extractive Summarization, Abstractive Summarization, Machine Learning, Youtube, Keyword Extraction.

LIST OF FIGURES:

1. INTRODUCTION:

1.1 INTRODUCTION:

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. 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 be very helpful because more useful information can be read in a short time. Generating summaries of transcripts is in the field of NLP because machines are required to understand what humans have written and produce human-readable outputs. Nowadays, it has become very easy to watch videos on Youtube for anything, be it educational or entertainment, there are lots of videos of those kinds of genres and because of the number, it is very important for us to find out the exact content that we want to consume. And sometimes because of network issues, we aren't able to watch text-based YouTube videos in high resolution, since it makes the text blurry. There are also those long and click bait videos that are only made to earn profit that contain not a single bit of useful information which wastes our time. So, by removing the useless part of the videos, skipping ads, and getting summaries, we can directly jump to the information that interests us and save lots of time and effort. There are countless situations when a user lures into the catchy title and thumbnail of the video and wastes all its time in consuming useless information. Students browse youtube videos before their exams so that they can get the most of the information in less time because of that they usually watch videos at double speed and because of this, it doubles the confusion about totally new topics. This is where our project is pointing out, it will create a summary from the transcript or the captions of whatever video you are watching. The most important part of our project is to collect the most necessary information and concentrate it into a small paragraph. Getting all the necessary information about the topic which interests you in the form of that small paragraph will save lots of time and effort for other things, which was wasted when you were lured into click bait videos and wasted your whole half or one hour or more.

1.2 MOTIVATION:

Many times, especially while searching for tutorial videos, we end up watching videos that aren’t relevant to us because of misleading titles, absence of description and clickbaits. The motivation of this project is to solve this issue by building an optimized and effective system.which will summarize any YouTube video with available transcripts. With the help of this summary, the user can determine the relevance of the video’s content without watching

the video. This will eventually save the time and energy of the user and increase work efficiency by filtering unsuitable videos.

1.3 PROBLEM STATEMENT AND OBJECTIVE:

PROBLEM STATEMENT:

300 hours of video are uploaded to YouTube every minute! Much of this content is verbose like tutorials, speeches, educational content. It is quite difficult to spend time watching such videos and sometimes our efforts may become futile if we couldn't find relevant information out of it. We have developed a Chrome extension that curbs this issue by summarizing the captions or transcript of any given video, it is able to pull the most important information and condense it into a small paragraph, reading this paragraph would take a tiny fraction of the total length of the video, while still providing the most important points to the user.

OBJECTIVES:

1. To generate Summary of YouTube video in the browser through a Chrome Extension.
2. To build a robust model which can generate summaries for videos on a wide range of topics.

2 LITERATURE SURVEY:

2.1 SURVEY OF EXISTING SYSTEM:

1. **Taskiran, Cüneyt & Amir, Arnon & Ponceleon, Dulce & Delp, Edward. (2002). Automated video summarization using speech transcripts. 4676. 371-382. 10.1117/12.451107.**

   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. We propose a method to automatically generate video summaries for long videos. Our video summarization approach involves mainly two tasks: first, segmenting the video into small, coherent segments and second, ranking the resulting segments. Our proposed algorithm scores segments based on word frequency analysis of speech transcripts. Then a summary is generated by selecting the segments with the highest score to duration ratios and these are concatenating them. We have designed and performed a user study to evaluate the quality of summaries generated. Comparisons are made using our proposed algorithm and a random segment selection scheme based on statistical analysis of the user study results. Finally we discuss various issues that arise

in evaluation of automatically generated video summaries**.**

2. **Taewon Yoo, Hyewon Jeong, Donghwan Lee, and Hyunggu Jung. 2021. LectYS: A System for Summarizing Lecture Videos on YouTube. In the 26th International Conference on Intelligent User Interfaces - Companion(IUI '21 Companion). Association for Computing Machinery, New York, NY, USA, 90–92. DOI:https://doi.org/10.1145/3397482.3450722**

   Students leverage online resources such as online classes and YouTube is increasing. Still, there remain challenges for students to easily find the right lecture video online at the right time. Multiple video search methods have been proposed, but to our knowledge, no previous study has proposed a system that summarizes YouTube lecture videos using subtitles. This demo proposes LectYS, a system for summarizing lecture videos on YouTube to support students search for lecture video content on YouTube. The key features of our proposed system are: (1) to summarize the lecture video using the subtitle of the video, (2) to access the specific parts of the video using the start time of video subtitle, and (3) to search for the video with keywords. Using LectYS, students are allowed to search for lecture videos on YouTube faster and more accurately.

3. **Song H-J, Kim H-K, Kim J-D, Park C-Y, Kim Y-S. Inter-Sentence Segmentation of YouTube Subtitles Using Long-Short Term Memory (LSTM).** ***Applied Sciences*****. 2019; 9(7):1504. https://doi.org/10.3390/app9071504**

   This article suggests that how the an Video description and textual summarization have been widely studied in recent years. In this section, we have discussed research work on traditional and then neural network based models on both problem areas. Finally, we have introduced a few related works with models that jointly deal with video description and textual summarization. YouTube, a video-sharing site, provides captions in many languages. Currently, the automatic caption system extracts voice data when uploading a video. We propose a method to divide text into sentences and generate period marks. This could improve the accuracy of automatic translation of English subtitles.

4. **Bokhove C, Downey C. Automated generation of 'good enough' transcripts as a first step to transcription of audio-recorded data. Methodological Innovations. 2018;11(2). doi:10.1177/2059799118790743**

   This article presents a proof-of-concept exploration utilizing three examples of automated transcription of audio recordings from different contexts; an interview, a public hearing and a classroom setting, and compares them against 'manual' transcription techniques in each case.
   It begins with an overview of literature on automated captioning and the use of voice

recognition tools for the purposes of transcription. Originality checking software was used to determine a percentage match between the automated transcript and a manual version as a basic measure of the potential usability of each of the automated transcripts. Some analysis of the more common and persistent mismatches observed between automated and manual transcripts is provided, revealing that the majority of mismatches would be easily identified and rectified in a review and edit of the automated transcript.These limitations notwithstanding, we conclude that this form of automated transcription provides 'good enough' transcription for first versions of transcripts.

2.2 LIMITATIONS OF EXISTING SYSTEMS:

In many of the plugins available, there is a limit on the number of tokens that can be processed (mostly 1024). Such plugins cannot generate summaries for long videos. This project overcomes this limitation.

2.3 MINIPROJECT CONTRIBUTION:

We distributed the entire project in subdivisions, mentioned below.

TEAM MEMBER 1 - Shriya Bijam:

- Researched and selected the problem statement.
- Built the Machine Learning part of the project.
- Fine-tuned the NLP model.
- Prepared the documentation and presentation for the project.

TEAM MEMBER 2 - Bharat Choudhary:

- Created API back-end using Flask
- Contributed in writing documentation for the project.

TEAM MEMBER 3 - Nikita Dubey :

- Created Front-End (Chrome extension) of the system,
- Prepared the documentation and presentation
- Maintained log book of this project.

3. PROPOSED SYSTEM:

We have created a chrome extension, which can be activated on any YouTube video whose transcript is available. The unique video id will then be used to fetch the video's transcript which will be passed to the ML model for generating summary. The summary will then be sent through HTTP in the form of JSON and will be displayed to the user. The basic strategy it uses is using ML summarizing techniques on the transcript of the video.

3.1 INTRODUCTION:

The project is divided into two separate entities:

1. **Client**
   It is a chrome extension which will render the summary of the youtube video below the youtube video player by making use of the above API. Just click on the summarize button and it will show the summary of the youtube video.

2. **Server**
   It is a simple flask app, which has an API /api/summarize?id=video_id which can be used to get the summary of a desired youtube video by simply making a GET XML HTTP request. The video id will then be used to fetch the video's transcript. With the help of distilbart-cnn-12-6 model, the summary of the transcript is generated and sent to the client as a JSON response.

3.2 ARCHITECTURE/FRAMEWORK:

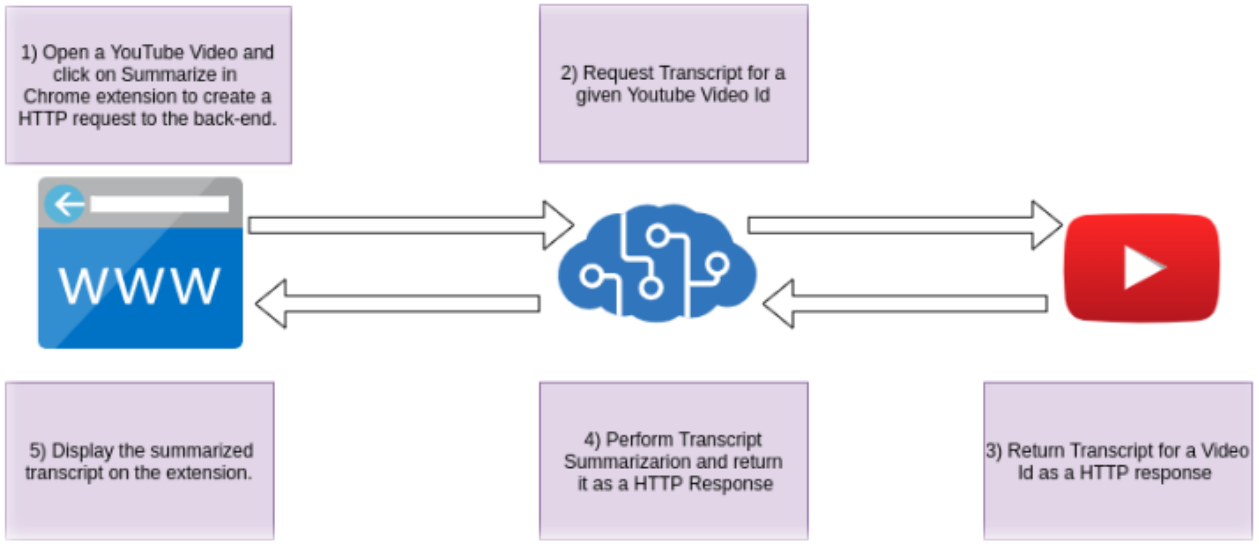

Fig 1: Block Diagram of Architecture.

3.3 ALGORITHM AND PROCESS DESIGN:

- **Get transcripts/subtitles for a given YouTube video Id.**

We utilize a python API called **youtube-transcript-api** which allows you to get the transcripts/subtitles for a given YouTube video. It also works for automatically generated subtitles, and supports translating subtitles. The API will return a list of dictionaries. We parse the data from the response to return the transcript in whole string format.

- **Perform text summarization on obtained transcripts using HuggingFace transformers.**

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:

1. Extractive Summarization: This is where the model identifies the important sentences and phrases from the original text and only outputs those. In this project, we will use transformers for this approach.
2. 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. In this step, we will use HuggingFace's transformers library in Python to perform extractive text summarization on the transcript obtained from previous step.

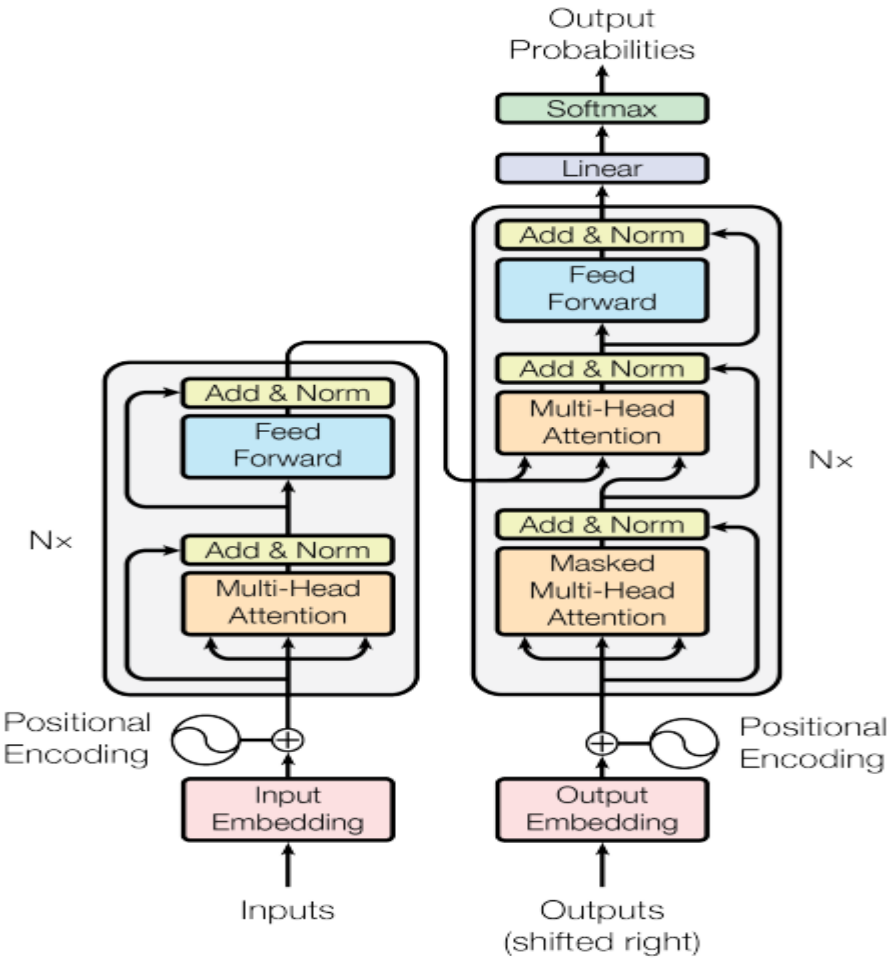

Fig 2: The Transformer-model architecture

- **Build a Flask backend REST API to expose the summarization service to the client.**

This is an extremely simple application, we only have a single endpoint, so our only resource will be the summarized text.

API endpoints: http://[hostname]/api/summarize?id=<video_id>

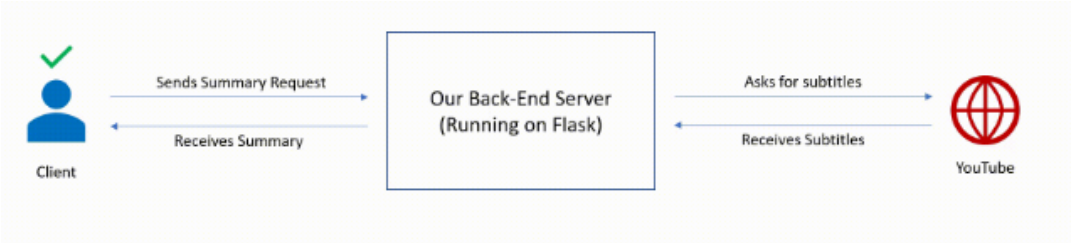

Fig 3: API Architecture

- **Develop a chrome extension which will utilize the backend API to display summarized text to the user.**

  On clicking the summarize button on the popup, if the url is of form *https://www.youtube.com/watch?v=** the popup js makes a GET request to our *Server API* .A div element is added below the youtube player with a preload text. Then, after the text is received, it is passed to the content js which then changes the content inside the above div element. Most of the properties are inherited from the parent element, so that it fits perfectly there. Extra styling is added in content.css. It can be used by loading unpacked from chrome://extensions/.

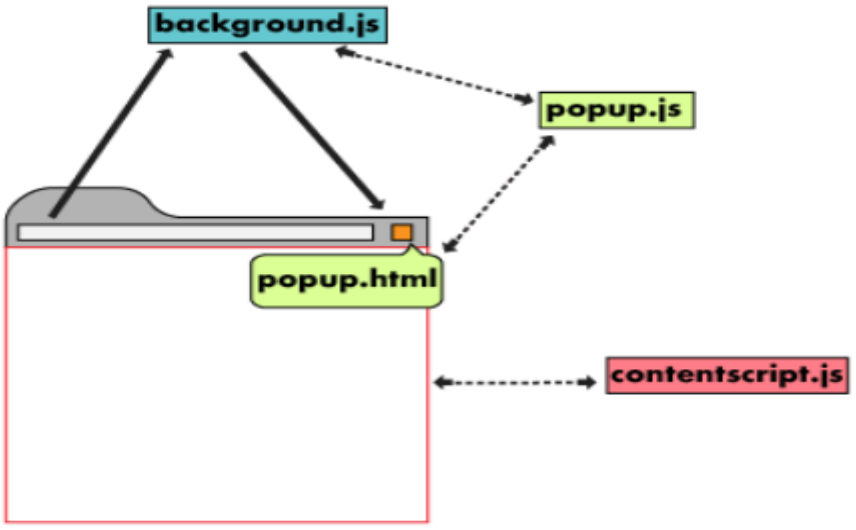

Fig 4: Role of each of the files for building a chrome extension.

3.4 DETAILS OF HARDWARE AND SOFTWARE:

Hardware:

- **Memory: 8 GB RAM**
- **Storage: 500 GB internal storage drive**

Software:

1. **Flask:** Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.
2. **youtube-transcript-api:** This is a python API which allows you to get the transcript/subtitles for a given YouTube video.
3. **PyTorch:** An open source machine learning framework that accelerates the path from research prototyping to production deployment.
4. **HuggingFace Transformers:** It provides APIs to quickly download and use those pretrained models on a given text, fine-tune them on our own datasets and then share them with the community.

3.5 EXPERIMENT AND RESULT:

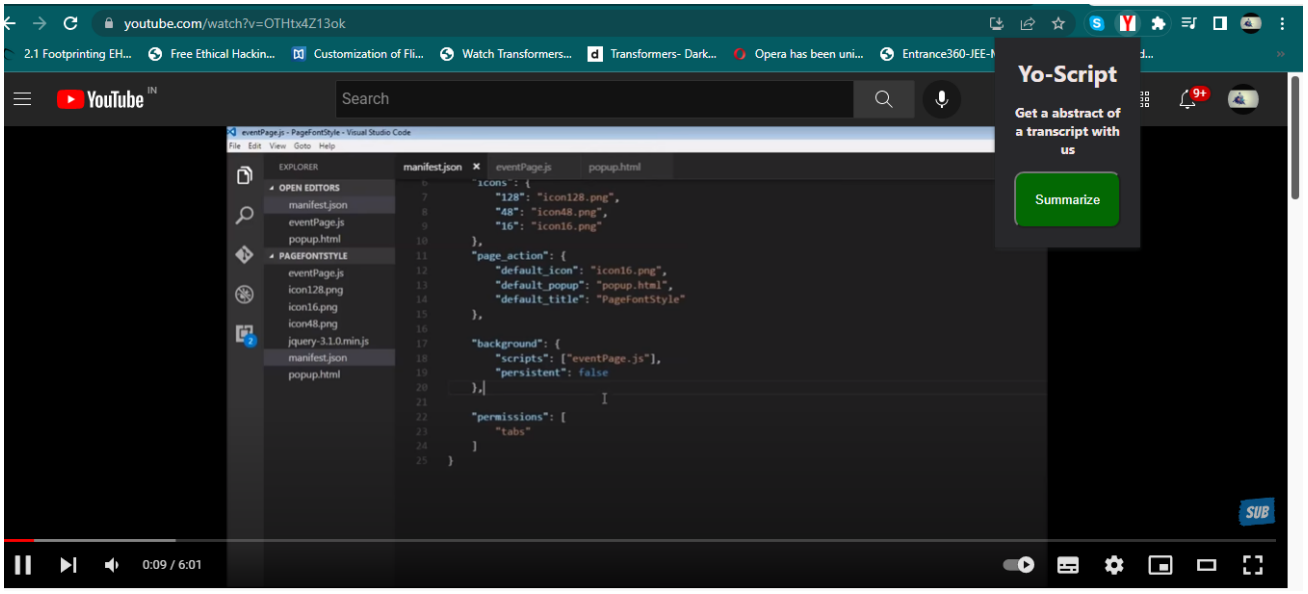

Fig 5: Initial page of the extension

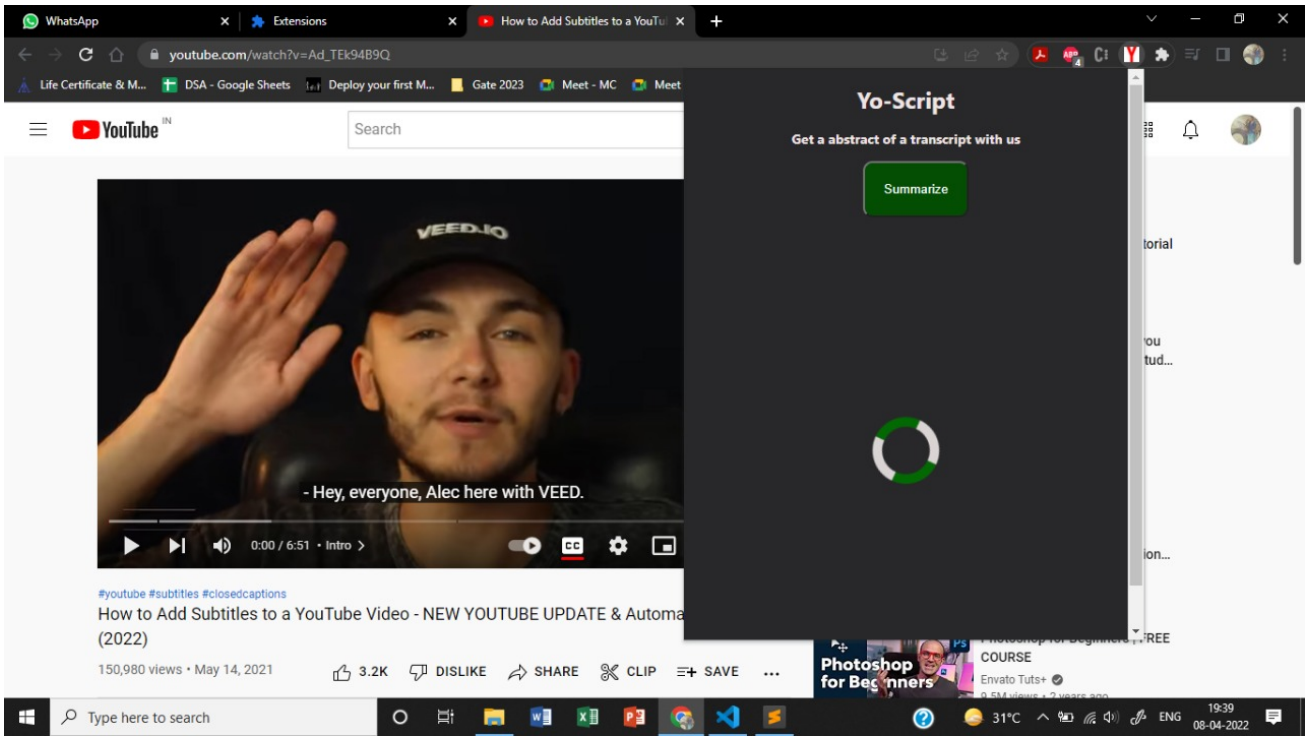

Fig 6: Loading screen of the extension

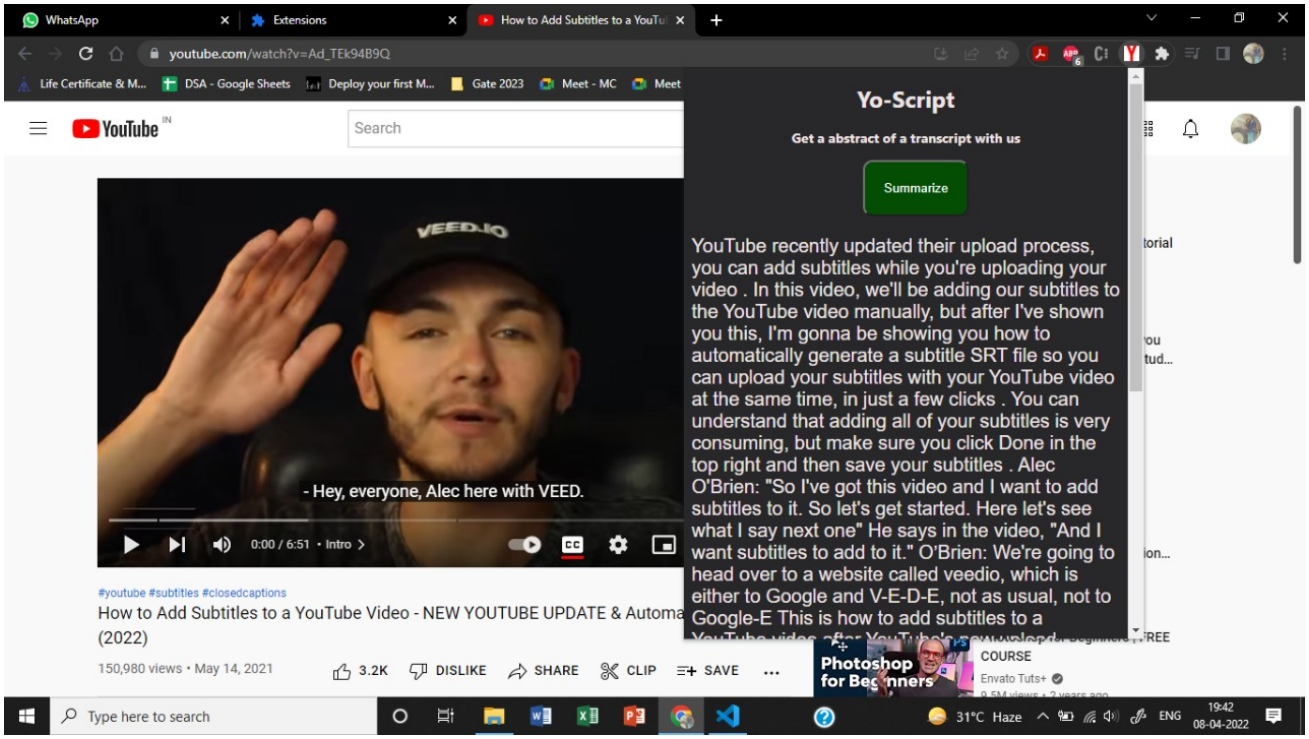

Fig 7: Final screen displaying summary of the video

3.6 CONCLUSION:

The goal of this summarizer is to reduce reading time, make the selection process easier, less biased than humans and present information in a way that is abbreviated and conserves the central content of the original article. This representation will not only save processing time, but will also save users from clicking baited videos. In this, we created a short and a fluent summary of a longer text transcript of a video, which mines appropriate information from the transcript to utilize the relevant information faster. Upon request from the user, this easy-to-use user interface would be able to display the text there only.

We are sure this project will leave users satisfied and will solve all of the problems that it's supposed to tackle i.e. saving your crucial and essential time, providing you the knowledge and data that you seek all the while making sure you can check out all the videos that important to you and not waste your precious time on one long video.

Future Work:

1. Presently, only basic setup is done. Exception handling is left, like: The video is live, url is not correct.
2. Summarization needs to be improved, if the content can be broken in some parts and then summarization is done and that is given as time stamps. Sumy can be used.
3. The transcript is presently working only for youtube videos having captions on, the audio and text processing are left.
4. There can be extra features of choosing language and length of summary. There can be an option to convert the text to audio .
5. The summary of some 10-15 minutes as time stamps can be given, then a person can also search for a keyword, when it is discussed and can go directly to that time .

REFERENCES:

1. https://medium.com/newolf-society/samuraiser-the-youtube-transcript-summarizing-extension-89434153d2bc
2. https://www.researchgate.net/publication/220979489_Automated_video_summarization_using_speech_transcripts
3. https://dl.acm.org/doi/10.1145/3397482.3450722
4. https://www.crio.do/projects/python-youtube-transcript/
5. https://betterprogramming.pub/the-ultimate-guide-to-building-a-chrome-extension-4c01834c63ec
6. https://www.irjmets.com/uploadedfiles/paper/volume3/issue_4_april_2021/7980/1628083331.pdf