Report On

YouTube Transcript Summarizer

Submitted in partial fulfillment of the requirements of the Course project
in Semester IV of Second Year Computer Engineering
by

Atharv Sawant (Roll No. 30)
Dastigeer Sayyed (Roll No. 34)
Shakib Shaikh (RollNo.36)

Supervisor

Prof. Sneha Mhatre

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(2023-24)
Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled “YouTube Transcript Summarizer” is a Bonafide work of Atharv Sawant(Roll No.30), Dastigeer Sayyed(Roll No.34) Shakib Shaikh(Roll No.36) submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester IV of Second Year Computer Engineering.

Supervisor

Prof. Sneha Mhatre

Internal Examiner

External Examiner

Dr Megha Trivedi
Head of Department

Dr. H.V. Vankudre
Principal

Index

1.Abstract

The "YouTube Transcript Summarizer" application is a Python-based graphical user interface (GUI) tool developed using the Tkinter library. Its primary function is to assist users in extracting and summarizing transcripts from YouTube videos. By leveraging the YouTube Transcript API, the application allows users to input a YouTube video link through a designated entry field. Upon submission, the application fetches the transcript of the corresponding video. Through natural language processing techniques facilitated by the NLTK library, the transcript text is tokenized into sentences and words. Stop words are removed to enhance the quality of the summary. The application then computes word frequencies to identify the most significant terms within the transcript. Utilizing these key terms, it selects sentences containing these terms to generate a concise summary. The summarized text is then presented to the user in a new window, featuring a

scrollable text widget for easy readability. Overall, the "YouTube Transcript Summarizer" empowers users with a streamlined and efficient means of extracting essential information from YouTube videos, catering to a diverse range of information retrieval needs.

2.Problem Statement

The "YouTube Transcript Summarizer" application revolves around the need for an efficient tool to extract and summarize transcripts from YouTube videos. As the volume of video content on platforms like YouTube continues to grow exponentially, users often encounter challenges in navigating and extracting relevant information from lengthy video transcripts. This poses a significant obstacle for individuals seeking specific information or insights within these videos. The lack of effective tools to summarize video content impedes users' ability to efficiently consume and utilize the wealth of information available on online video platforms. Therefore, the problem statement addresses the necessity for a user-friendly application that can seamlessly retrieve, process, and present summarized transcripts from YouTube videos, offering users a convenient solution to extract key insights and information from video content.

3. Module Description

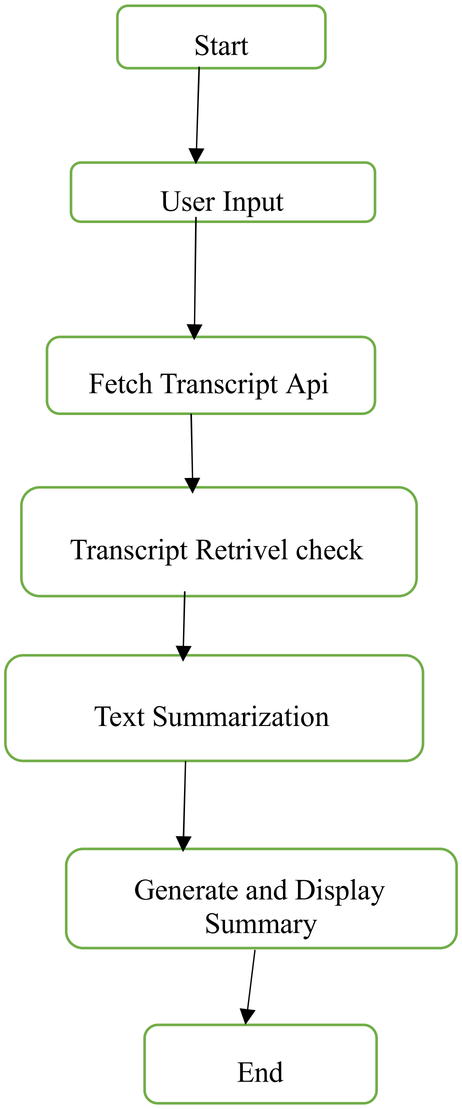
Tkinter Module: This module is responsible for creating the graphical user interface (GUI) of the application. It utilizes Tkinter library functionalities to design windows, labels, entry fields, buttons, and other GUI components necessary for user interaction.

PIL Module: The PIL (Python Imaging Library) module is employed for image processing tasks within the GUI. It allows the application to display background images seamlessly, enhancing the visual appeal of the user interface.

YouTube Transcript API Module: This module interfaces with the YouTube Transcript API to fetch transcripts from YouTube videos. It facilitates the extraction of textual content from the videos, which serves as the input for the summarization process.

NLTK Module: The NLTK (Natural Language Toolkit) module implements natural language processing techniques for text summarization. It tokenizes the transcript text into sentences and words, removes stop words to improve summary quality, and identifies key terms using word frequency analysis. Additionally, it selects relevant sentences containing these key terms to generate a concise summary of the video transcript.

4.Block Diagram



5.Code

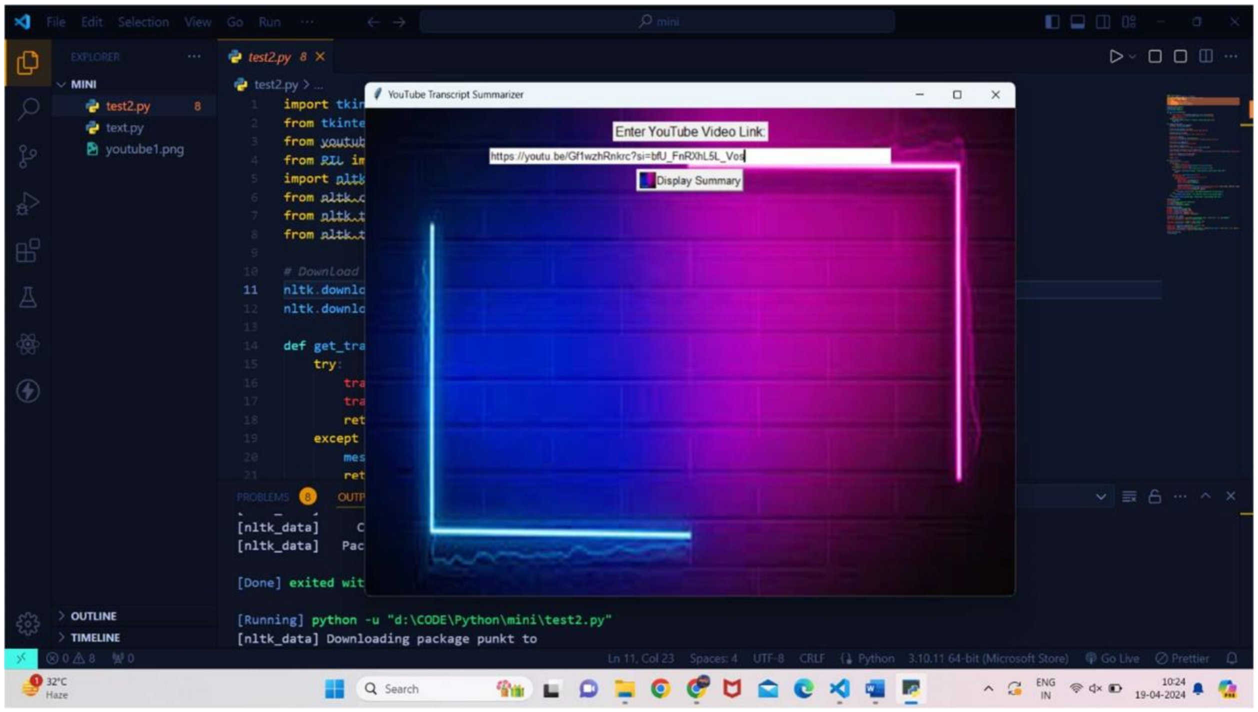
```
import tkinter as tk
from tkinter import messagebox, scrolledtext from
youtube_transcript_api import YouTubeTranscriptApi from
PIL import Image, ImageTk import nltk
from nltk.corpus import stopwords from
nltk.tokenize import word_tokenize from
nltk.tokenize import sent_tokenize

# Download NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
```

```
def get_transcript(video_id):
try:
        transcript_list = YouTubeTranscriptApi.get_transcript(video_id)
transcript_text = ' '.join([t['text'] for t in transcript_list])         return
transcript_text     except Exception as e:
        messagebox.showerror("Error", f"Failed to fetch transcript.\n{e}")
return None

def   summarize_text(text):             #
Tokenize   the   text   into   sentences
sentences = sent_tokenize(text)

    # Tokenize each sentence into words
    words = [word_tokenize(sentence) for sentence in sentences]

    # Flatten the list of words
    words = [word for sublist in words for word in sublist]

    # Filter out stopwords
    stop_words = set(stopwords.words('english'))
    filtered_words = [word for word in words if word.lower() not in stop_words]

    # Calculate word frequencies
word_freq = nltk.FreqDist(filtered_words)

    # Get the most frequent words
    top_words = [word for word, freq in word_freq.most_common(10)]

    # Get sentences containing the most frequent words
    summary_sentences = [sentence for sentence in sentences if any(word in sentence.lower() for
word in top_words)]

    # Join the sentences to form the summary
summary = ' '.join(summary_sentences)
```

```
    return summary


def display_summary():
    video_link = video_entry.get()
if video_link:
        if "watch?v=" in video_link:
            video_id = video_link.split("watch?v=")[-1].split("&")[0]
elif "youtu.be" in video_link:
            video_id = video_link.split("youtu.be/")[-1].split("?")[0]
else:
            messagebox.showwarning("Warning", "Please provide a valid YouTube video link.")
return


        transcript_text = get_transcript(video_id)
if transcript_text:
            summarized_text = summarize_text(transcript_text)
if summarized_text:
                summary_window = tk.Toplevel(root)
                summary_window.title("Summary")
summary_window.geometry("800x600")


                # Display summarized text
                summary_text_display = scrolledtext.ScrolledText(summary_window, wrap=tk.WORD,
width=100, height=30)
                summary_text_display.insert(tk.END, summarized_text)
summary_text_display.pack(padx=20, pady=20)             else:
                messagebox.showinfo("Info", "No summary generated for the transcript.")
else:
            messagebox.showinfo("Info", "No transcript found for the provided video.")
else:
        messagebox.showwarning("Warning", "Please enter a valid YouTube video link.")


# Create main window root
= tk.Tk()
```

```
root.title("YouTube Transcript Summarizer")
root.geometry("800x600")
root.configure(background='#f0f0f0')

# Load background image bg_image =
Image.open("youtube1.png") bg_image =
bg_image.resize((800, 600)) bg_image =
ImageTk.PhotoImage(bg_image) bg_label =
tk.Label(root, image=bg_image)
bg_label.place(x=0, y=0, relwidth=1,
relheight=1)

# Create and place widgets
video_label = tk.Label(root, text="Enter YouTube Video Link:", font=("Arial", 12), bg="#f0f0f0")
video_label.place(relx=0.5, rely=0.05, anchor="center")

video_entry = tk.Entry(root, width=70, font=("Arial", 10))
video_entry.place(relx=0.5, rely=0.1, anchor="center")

display_icon = Image.open("youtube1.png").resize((20, 20)) display_icon
= ImageTk.PhotoImage(display_icon)
display_button = tk.Button(root, text="Display Summary", command=display_summary,
font=("Arial", 10), image=display_icon, compound="left")
display_button.place(relx=0.5, rely=0.15, anchor="center")

# Start GUI main loop root.mainloop()
```

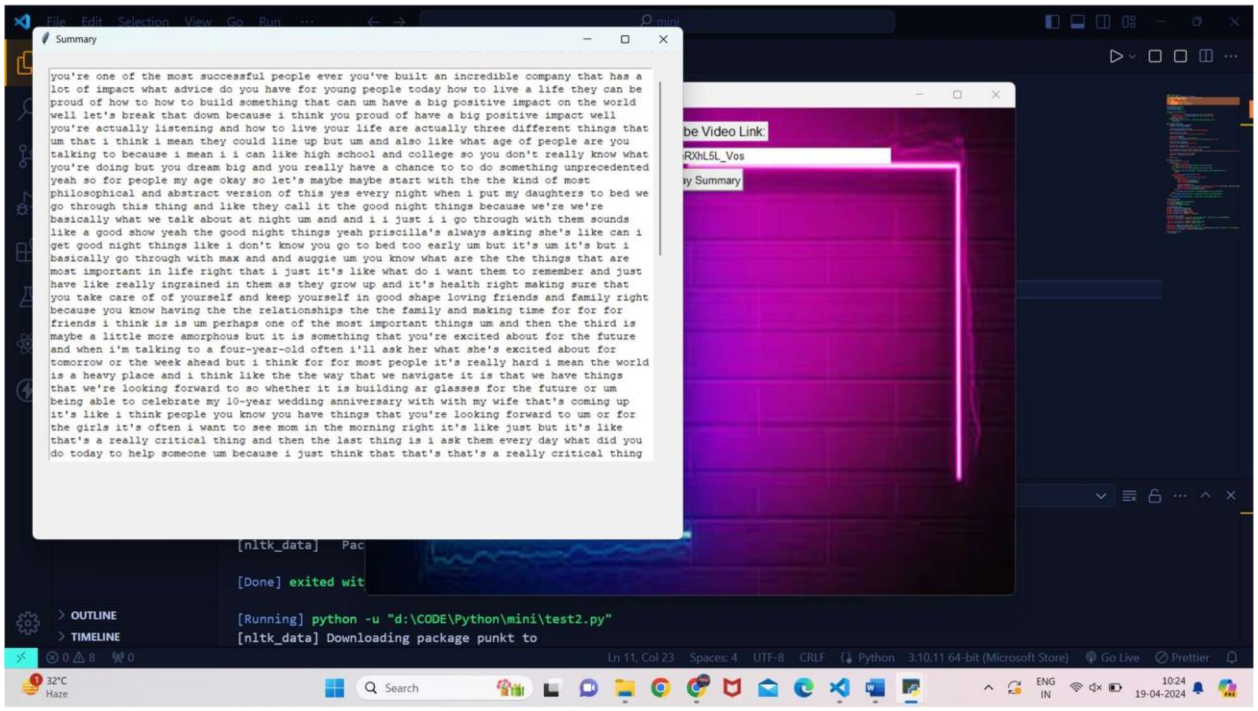
6.Results



```
https://youtu.be/Gf1wzhRnkrc?si=bfU_FnRXhL5L_Vos
```

```
import tkin
from tkinte
from youtub
from PIL im
import nltk
from nltk.c
from nltk.t
from nltk.t

# Download
nltk.downlo
nltk.downlo

def get_tra
    try:
        tra
        tra
        ret
    except
        mes
        ret
```

```
[nltk_data]     C
[nltk_data]   Pac

[Done] exited wit

[Running] python -u "d:\CODE\Python\mini\test2.py"
[nltk_data] Downloading package punkt to
```



```
you're one of the most successful people ever you've built an incredible company that has a
lot of impact what advice do you have for young people today how to live a life they can be
proud of how to how to build something that can um have a big positive impact on the world
well let's break that down because i think you proud of have a big positive impact well
you're actually listening and how to live your life are actually three different things that
um that i think i mean they could line up but um and also like what age of people are you
talking to because i mean i i can like high school and college so you don't really know what
you're doing but you dream big and you really have a chance to to do something unprecedented
yeah so for people my age okay so let's maybe maybe start with the the kind of most
philosophical and abstract version of this yes every night when i put my daughters to bed we
go through this thing and like they call it the good night things because we're we're
basically what we talk about at night um and and i i just i i go through with them sounds
like a good show yeah the good night things yeah priscilla's always asking she's like can i
get good night things like i don't know you go to bed too early um but it's um it's but i
basically go through with max and and auggie um you know what are the the things that are
most important in life right that i just it's like what do i want them to remember and just
have like really ingrained in them as they grow up and it's health right making sure that
you take care of of yourself and keep yourself in good shape loving friends and family right
because you know having the the relationships the the family and making time for for for
friends i think is is um perhaps one of the most important things um and then the third is
maybe a little more amorphous but it is something that you're excited about for the future
and when i'm talking to a four-year-old often i'll ask her what she's excited about for
tomorrow or the week ahead but i think for for most people it's really hard i mean the world
is a heavy place and i think like the the way that we navigate it is that we have things
that we're looking forward to so whether it is building ar glasses for the future or um
being able to celebrate my 10-year wedding anniversary with with my wife that's coming up
it's like i think people you know you have things that you're looking forward to um or for
the girls it's often i want to see mom in the morning right it's like just but it's like
that's a really critical thing and then the last thing is i ask them every day what did you
do today to help someone um because i just think that that's that's a really critical thing
```

7.Conclusion

The "YouTube Transcript Summarizer" application presents an effective solution for extracting and summarizing textual content from YouTube videos. By leveraging a combination of Tkinter for GUI development, PIL for image processing, the YouTube Transcript API for transcript retrieval, and NLTK for text summarization, the application offers users a seamless and intuitive interface to access key insights and information within video transcripts. Through its summarization process, the application enhances user productivity by condensing lengthy transcripts into concise summaries, enabling efficient consumption of video content. Overall, the "YouTube Transcript Summarizer" serves as a valuable tool for users seeking to extract meaningful insights and information from YouTube videos, enhancing their ability to access and utilize the vast array of content available on online video platforms.

8.References

Stackoverflow,
tutorialpoints ,
github,
Openai