VID TRANSCRIPT SUMMARIZER

A Mini Project Report

Submitted by

Y.Deekshita (21331A12C6)

P.Vijay sri harsha (21331A1285)

T.Jahnavi (21331A12B2)

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

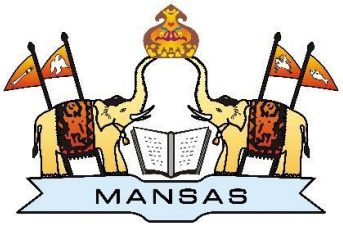
IN

INFORMATION TECHNOLOGY

Under the esteemed Guidance of

Mrs.P.Ramya

(Assistant Professor)



DEPARTMENT OF INFORMATION TECHNOLOGY

MVGR COLLEGE OF ENGINEERING (A),

VIZIANAGRAM.

April 2024

DECLARATION

We hereby declare that the mini project entitled "**VID TRANSCRIPT SUMMARIZER"** submitted for the partial fulfilment of B.Tech Degree is our original work and the project has not formed the basis for the award of any degree or any other similar titles.

Place: Vizianagaram

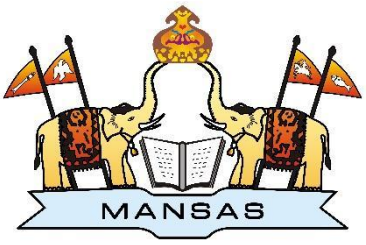
Date:

Y.Deekshitha(21331A12C6)

T.Jahnavi (21331A12B2)

P.Vijay sri harsha (21331A1285)

CERTIFICATE



This is to certify that the project entitled "**VID TRANSCRIPT SUMMARIZER**" is the bonafide work carried out by P.Vijay sri harsha (21331A1285), T.Jahnavi(21331A12B2), Y.Deekshita (21331A12C6), students of B.Tech VI semester, Information Technology, MVGR College of Engineering (Autonomous), Vizianagaram, during the year 2020-2024, in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology and that the project has not formed the basis for the award previously of any degree or any other similar title.

Signature of MiniProject Guide

Mrs.P.Ramya,

(Assistant Professor),

Department of IT,

MVGR College of Engineering (A).

Signature of Head of the department

Dr. P Sreenivasa Rao, M.Tech., Ph.D.,

Professor & Head,

Department of IT,

MVGR College of Engineering (A).

ACKNOWLEDGMENT

We wish to express our sincerest and most profound gratitude to our guide Mrs.P.Ramya(Assistant Professor), Department of Information Technology. We all thankful for his cooperation and guidance. She had given all the valuable instructions and support, even at odd hours and gave her precious time.

We thank our Head of the Department, Dr. P Sreenivasa Rao, Professor, Department of Information Technology for providing us all the necessary infrastructure and support whenever necessary.

We also thank Dr. R Ramesh, Principal of MVGR college of Engineering, for extending his utmost support and cooperation in providing all provisions for successful completion of the project.

We sincerely thank the staff of Department of Information Technology for helping us out in all the ways we needed and they could have.

Y.Deekshitha(21331A12C6)

T.Jahnavi (21331A12B2)

P.Vijay sri harsha (21331A1285)

ABSTRACT

Enormous number of recordings are being created and uploaded in the internet everyday.Over millions of user are using youtube in a day but the main problem is that the video recordings subtitles are probably too big for the person to read so we using this vid transcripter and summarizer one can summarize the content of the particular video.

For this the solution we adapted the first transcription of the video and summarized using the languages like python for implementation of the youtube transcripter and summarizer we can also define the specific lengths for the vid transcripter and summarizer.

It helps us to save time. Instead of using a lot of time reading the transcripters we summarize the transcripts which helps us to save your time and also save your memory.

PROGRAM OUTCOMES

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the

professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Title	P.O Mapping
Vid-transcript summarizer	1,2,3,4,5,9,10,11,12

Table of Contents

1 . INTRODUCTION

1.1 PROJECT OVERVIEW

This YouTube Transcripter Summarizer is a tool designed to extract key insights and summarize the contents of YouTube videos through their transcripts.

This innovative technology utilizes natural language processing (NLP) algorithms to analyze the spoken content of videos and condense it into concise summaries.

Nowadays most students use youtube to learn things because they won’t listen to classes or maybe by any difficulty a student has missed the class then due to any reason he can’t get to a particular faculty .

So he uses any online learning platform if the student doesn't have enough time to watch the full video .So, we decided to create a solution for the problem facing by the students or any one who has less time to make a short note.

This vid transcripter summarizer saves your time and also your memory.

Firstly it transcripts the text and summarize it later for efficient use

of the vid transcript summarizer.

1.2 LITERATURE SURVEY

REF. NO:	AUTHOR	YEAR	TITLE	DESCRIPTION	TECHNIQUE	METRICS USED:	OBSERVATIONS
1	Nallapa	2018	Youtube Transcript Summarizer	Videos which may have a longer duration converted into summarized text.	Using NLP	BERT Score, Semantic similarity	Summarizing transcripts of videos automatically into important patterns.
2	Janipatel	2023	Text summarizatio n using NLP	In these text into sentences and using matrix converted to summary.	Using NLP	F1 score, Semantic similarity	In these he used hindi scripts for summary.
3	Gousiya Begum	2022	Youtube transcript summarizer	Videos which may have a longer duration converted into summarized text.	Web API	ROUGE	Long videos Which will take more time are summarized.
4	S. Tharun	2022	Transcript Summarizatio n of YouTube Videos	Constructing a brief and fluent summary while keeping vital information and overall meaning.	REST API, NLP	Readabilit y Metrics, User Evaluation	Summarizing transcripts of videos automatically into fluent summary.
5	Aniqa Dilawari	2019	Abstractive Summarizatio n Of Video Sequences,	This provides a text-based video description and abstractive summary	Using NLP	F1 score, Semantic similarity	In these he used hindi scripts for summary.
6	Pradeep, Soumya	2017	Real time video summarizatio n On mobile platform	This technique uses the analyzing into the video stream	Using deep learning	Histogram	It gives an easy idea to look up important content.
7	R. Kranthi Kumar	2022	Transcript Summarizatio n of YouTube Videos	Constructing a brief and fluent summary while keeping vital information and overall meaning.	REST API, NLP	Readabilit y Metrics, User Evaluation	Summarizing transcripts of videos automatically into a fluent summary.
8	Eesha Inamdar	2023	YOUTUBE TRANSCRIP T SUMMARIZ ER	This provides a text-based video description and abstractive summary	Pytube, Flask	ROUGE	To improve surfing experience without being distracted.
9	Prashu Pandey	2023	Transcript Summarizer	In these text into sentences and using matrix converted to summary.	Flask API, Web API	Semantic Similarity	It gives an easy idea to lookup important content.

1.3 EXISTING SYSTEM

The existing YouTube transcript summarizer system is designed to process video transcripts using Natural Language Processing (NLP) techniques. It starts by extracting the transcript from the YouTube video and then preprocesses the text data to remove noise and irrelevant information. The system utilizes various NLP libraries and algorithms to analyze the transcript and identify key themes, topics, and important sentences.

One of the key components of the system is the summarization algorithm, which generates a concise summary of the transcript. This algorithm considers factors such as sentence relevance, importance, and diversity to ensure that the summary captures the essence of the video. The system also includes a keyword extraction module to identify important keywords and phrases in the transcript.

To enhance the user experience, the system may include a user interface that allows users to interact with the summarized transcript. This interface may provide features such as highlighting important sentences, displaying key themes, and allowing users to navigate the summary easily.

Overall, the existing YouTube transcript summarizer system is designed to provide users with a quick and efficient way to understand the content of a video without having to watch the entire video. It leverages NLP techniques to process and summarize video transcripts, making it a valuable tool for content creators, researchers, and anyone looking to quickly grasp the key points of a YouTube video.

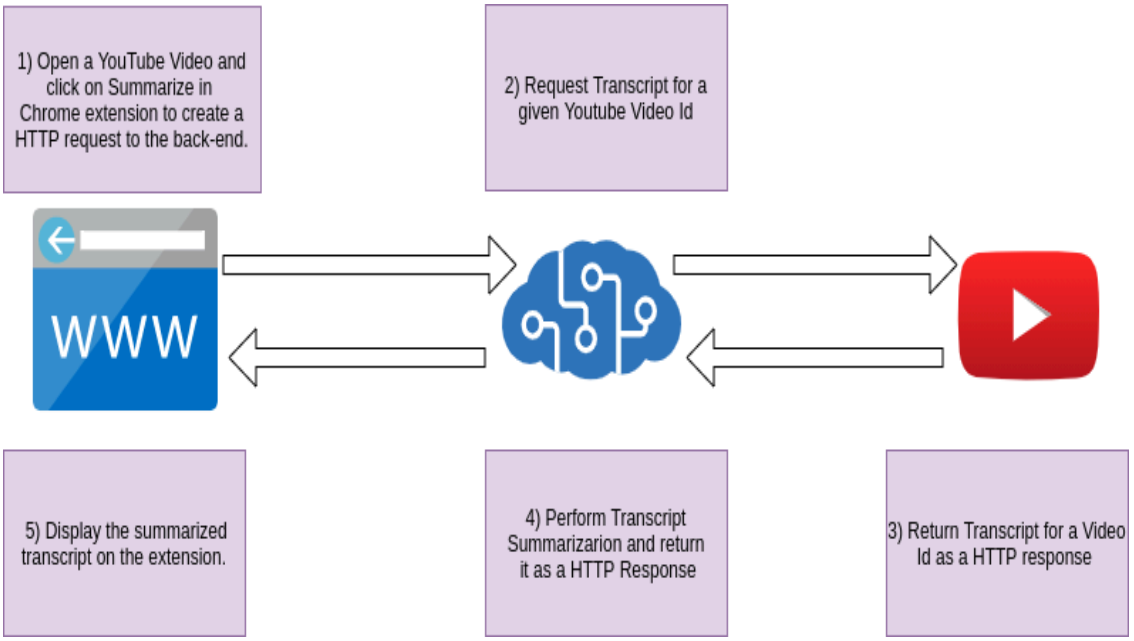
1.4 PROPOSED SYSTEM

The proposed video transcript Summarizer system utilizes youtube transcript api by importing YouTubeTranscriptApi we extract the youtube transcript from the videos then we convert the audio into text, followed by text preprocessing for cleaning and preparation. Then we use Streamlit it is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps.

Then we use google-generativeai for summarization of the processed text from the youtube transcript api and getting the thumbnail of the particular video .To reads key-value pairs from .env file and can set them as environment variables we used python-dotenv. It helps in the development of our application.Finally we used pathlib to direct all the paths of the libraries we used in the environment.

We then set the maximum and minimum length we wanted to have in your transcripts so the person may use it efficiently.This helps to save your time. We can also convert into many languages we prefer so it can be understandable for the viewer.

2. SYSTEM STUDY AND ANALYSIS



The project follows a clear flowchart as shown in Figure 1. Firstly, the user opens a YouTube video and clicks on the "summarize" button in the chrome extension. This initiates a HTTP request to the back-end of the system. Subsequently, the request is made to access the transcripts using the YouTube video ID obtained from the URL.

The response to this request will be a transcript of the video in single page summary format. Once the transcripts are obtained in text format, the system performs transcript summarization, which involves reducing the length of the transcript while retaining the most important information. Finally, the summarized transcript is displayed on the extension.

2.1 Functional Requirements

This project can be divided into 3 modules:

- importing
- transcription
- Generating

Importing:

- **Input**:The required files are imported like youtube transcript api,streamlit,pathlib.
  python_dotenv etc
- **Processing**: We process it in the terminal.
- **Output**: All the packages and files are downloaded.

Transcription:

- **Input**: Using the youtubetranscriptapi we download the transcripts.
- **Processing**: Running the transcripts.
- **Output**: The transcripts are found.

Generating:

- **Input**: using the functions the transcripts are summarized.
- **Processing**: the transcript is summarized.
- **Output**: A structured format(table) of the report is shown .

2.2 Non-Functional Requirements

- **Usability**:The interface is consistent and easy to operate with simple actions and it can be used by any no. of users.
- **Capability**:It is capable of holding many number of assignment files.
- **Performance**:Response time is minimal and latency in loading and running the application is very less
- **Availability**:It is available for all the users until the server is on and connected to the network.
- **Maintainability**:It is easy to use, maintain and add new features to it.

2.3 SYSTEM SPECIFICATIONS

2.3.1 HARDWARE SPECIFICATIONS

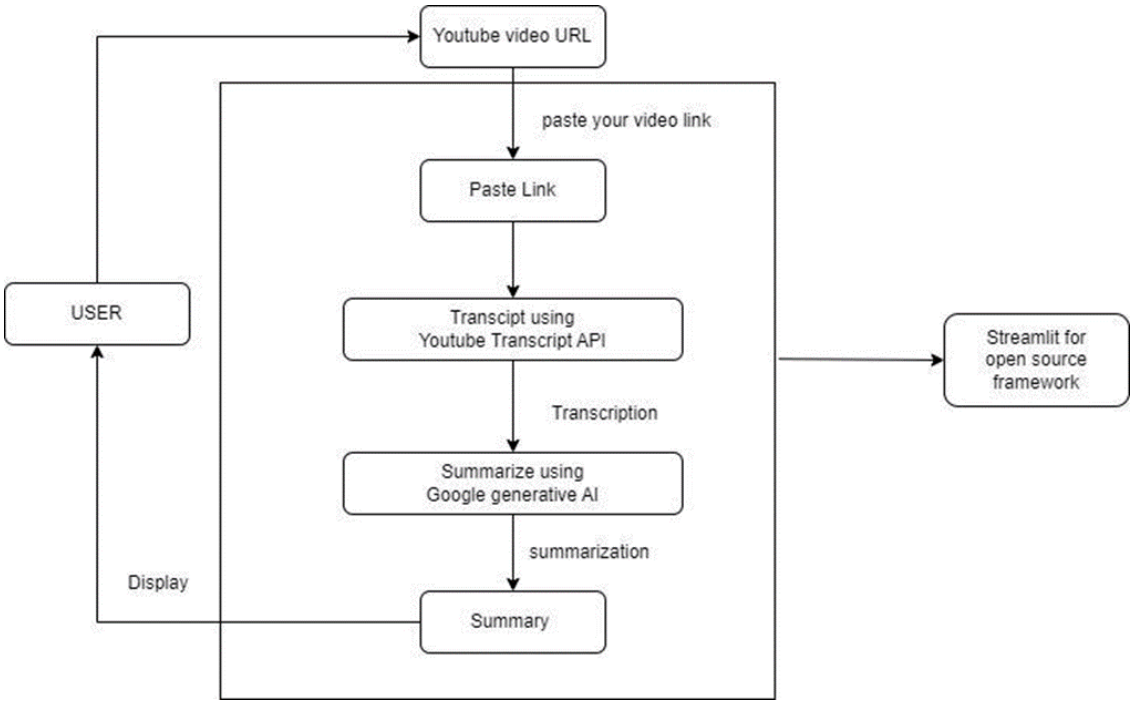
- MEMORY : As Transformers like bart need sufficient memory.
- CPU:For CPU acceleration.
- Pretrained models: pretrained models like bart are needed for the project.
- Storage: For storing the data sufficient storage is needed.

2.3.2 SOFTWARE SPECIFICATIONS

- Youtube transcript api
- streamlit
- google-generativeai
- python-dotenv
- Pathlib
- Anaconda

3.System Diagram

3.1 FLOW DIAGRAM:



4. CODE MODULE

Environmental variable:

GOOGLE-API-KEY="AIzaSyDkx4uepVsoQrL34Uq8AU__7fH3iC7l1p0
"

Requirements.txt:

youtube_transcript_api

streamlit

google-generativeai

python-dotenv

pathlib

google-cloud-translate

app.py:

```
import streamlit as st

from dotenv import load_dotenv

load_dotenv() ##load all the nevironment variables

import os

import google.generativeai as genai

from youtube_transcript_api import YouTubeTranscriptApi

genai.configure(api_key=os.getenv("GOOGLE-API-KEY"))
```

```
prompt="""You are a Youtube video summarizer. You will be taking the transcript text

and summarizing the entire video and providing the important summary in points

within 250 words. Please provide the summary of the text given here:  """

## getting the transcript data from yt videos

def extract_transcript_details(youtube_video_url):

    try:

        video_id=youtube_video_url.split("=")[1]

        transcript_text=YouTubeTranscriptApi.get_transcript(video_id)

        transcript = ""

        for i in transcript_text:

            transcript += " " + i["text"]

           return transcript

 except Exception as e:

        raise e

  def truncate_text(text, max_length, min_length):

    if len(text) <= max_length:

        return text

    elif len(text)<min_length:
```

```
        return text
    else:
        return text[:max_length-3] + '...'
## getting the summary based on Prompt from Google Gemini Pro
def generate_gemini_content(transcript_text,prompt):
 model=genai.GenerativeModel("gemini-pro")
    response=model.generate_content(prompt+transcript_text)
    return response.text
st.title("YouTube Transcript to Detailed Notes Converter")
youtube_link = st.text_input("Enter YouTube Video Link:")

if youtube_link:
    video_id = youtube_link.split("=")[1]
    print(video_id)
st.image(f"http://img.youtube.com/vi/{video_id}/0.jpg",
use_column_width=True)
if st.button("Get Detailed Notes"):
    transcript_text=extract_transcript_details(youtube_link)
    if transcript_text:
        summary=generate_gemini_content(transcript_text,prompt)
```

```
        max_summary_length = 300

        min_summary_length=250

        truncated_summary = truncate_text(summary, max_summary_length,
min_summary_length)

        st.markdown("## Detailed Notes:")

        st.write(truncated_summary)
```

5. TESTING & DEPLOYMENT

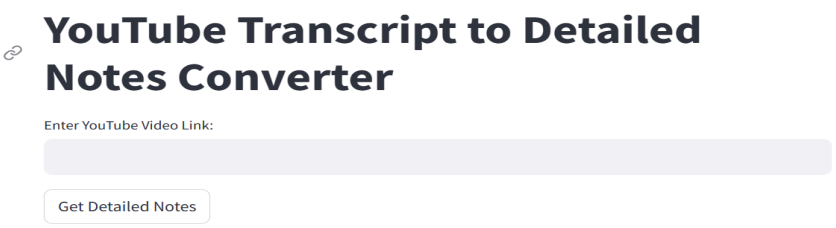
5.1 DEPLOYMENT STEPS:

1. First install Anaconda latest version and Python(3.10 version).
2. create an environment and get a key from Google API key from google AI studio.Set the environmental variable paths.
3. Now create three files with file name requirements.txt ,app.py and Environment variable.
4. Now paste the key in the Environment variable.
5. After add the library names required in the project they are
    - youtube_transcript_api
    - streamlit
    - google-generativeai
    - python-dotenv
    - pathlib
    - google-cloud-translate

6. Using “pip install -r requirements.txt” download the libraries.
7. Now write the code in the App.py and to run use streamlit run app.py
8. Finally run the folder in the visual studio to make sure that all the files created are present in a folder.
9. Paste the link in the streamlit Interface.

`

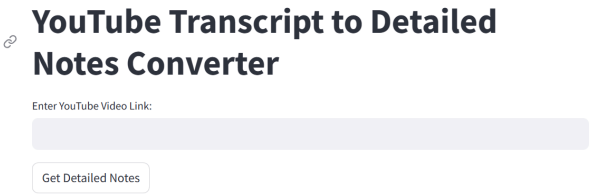
5.2. OUTPUT



5.3:TEST CASES:

- **TEST CASE-1**

First step:



Second Step:

Third step:

Summary:

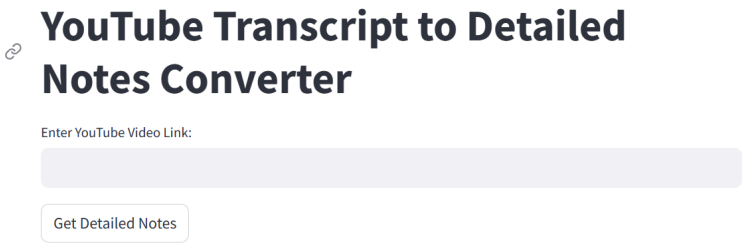
Cloud computing offers several advantages over on-premise computing for business expansion. It provides scalability, cost-effectiveness, enhanced data security, and reduced maintenance costs.

Key Points:

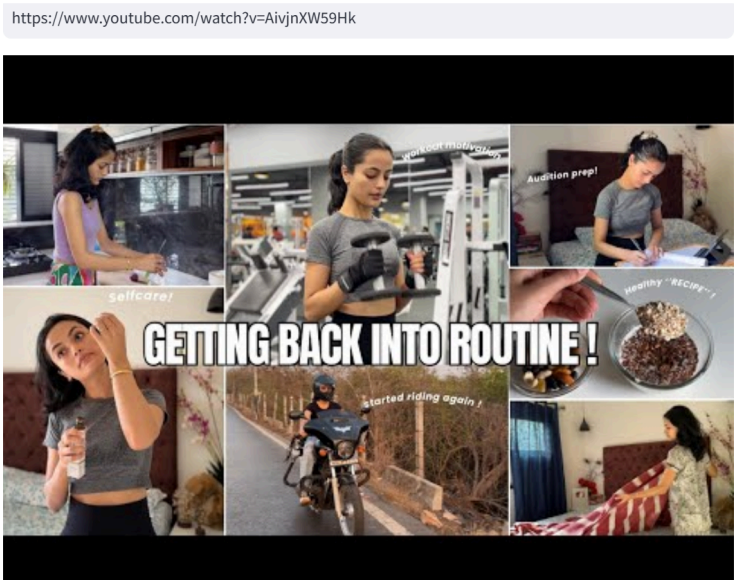
- **Scalability:** Cloud computing allows for easy up or down scaling, while on-premise setups have limited scalability options.
- **Cost:** Cloud computing involves paying only for used resources, while on-premise systems require substantial upfront investment in infrastructure and maintenance.
- **Data Security:** Cloud systems offer robust security measures and disaster recovery, whereas on-premise systems have limited data protection capabilities.
- **Maintenance:** Cloud service providers handle hardware and software maintenance, reducing the burden on business resources.
- **Deployment Models:** Cloud computing offers public, private, and hybrid cloud models, providing flexibility in cloud access.
- **Service Models:** Infrastructure as a Service (IaaS), Pl...

- **TEST CASE-2**

First step:



second step:



Third step:

6. CONCLUSION

6.1 Conclusion and Future scope

We proposed 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 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.

The field of YouTube transcript summarization is still evolving, and there is a lot of potential for future research and development. Some possible future scope for YouTube transcript summarizers are

1 Multi- lingual support

2 Real-time summarization

3 Incorporating user feedback

4 Image and video analysis

7. REFERENCES

INSTALLATIONS

YoutubeTranscriptApi

pip install youtube-transcript-api

Streamlit

pip install streamlit

Python-dotenv:

pip install python-dotenv

Anaconda:

https://www.anaconda.com/code-in-the-cloud.

PathLib:

pip install pathlib

RESOURCE

YOUTUBE :

"https://www.youtube.com/watch".