YOUTUBE TRANSCRIPT SUMMARIZER

A Summer internship Report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN
COMPUTER SCIENCE AND ENGINEERING-DATA SCIENCE

Submitted by

Mr. B. Maneesh Kumar (21071A6707)

Miss. K. Sai Praneetha (21071A6726)

Miss. K. Sai Sahithi (21071A6727)

Mr. T. Sathwik (21071A6753)

Under the guidance of

ASHALATHA.G
Assistant Professor
Department of Computer Science & Engineering- CSE-(CyS, DS) and AI&DS

DEPARTMENT OF CSE-(CYS, DS) and AI&DS

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT, AME B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

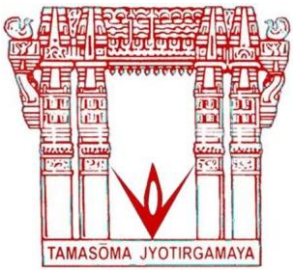
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT, AME B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF CSE-(CYS, DS) and AI&DS



CERTIFICATE

This is to certify that the project report entitled "**YouTube Transcript Summarizer"** is a bonafide work done under our supervision and is being submitted by **Mr. B. Maneesh Kumar(21071A6707), Miss. K. Sai Praneetha(21071A6726), Miss. K. Sai Sahithi (21071A6727), Mr. T. Sathwik (21071A6753)** in partial fulfillment for the award of the degree of Bachelor of Technology in CSE-CYS, DS and AI&DS , of the VNRVJIET, Hyderabad during the academic year 2023-2024. Certified further that to the best of our knowledge the work presented in this thesis has not been submitted to any other University or Institute for the award of any Degree or Diploma.

Ashalatha. G
Assistant Professor
Dept. of CSE-(CYS, DS) and AI&DS
VNR VJIET

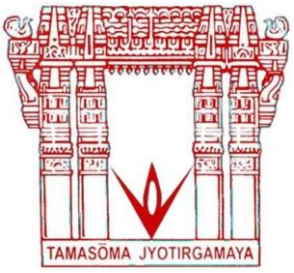
Dr. M. Raja Sekar
Professor and Head
Dept. of CSE-(CYS, DS) and AI&DS
VNR VJIET

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT, AME B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF CSE-(CYS, DS) and AI&DS



DECLARATION

We declare that the major project work entitled "**YouTube Transcript Summarizer**" submitted in the department of **CSE-(CYS, DS) and AI&DS**, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **CSE-(CYS, DS) and AI&DS** is a bonafide record of our own work carried out under the supervision of Ashalatha.G**, Assistant Professor, Department of CSE-(CYS, DS) and AI&DS, VNRVJIET**. Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad

B. Maneesh Kumar	**K. Sai Praneetha**	**K. Sai Sahithi**	**T. Sathwik**
(21071A6707)	(21071A6726)	(21071A6727)	(21071A6753)

ACKNOWLEDGEMENT

Firstly, we would like to express our immense gratitude towards our institution VNR Vignana Jyothi Institute of Engineering and Technology, which created a great platform to attain profound technical skills in the field of Computer Science, thereby fulfilling our most cherished goal.

We are very much thankful to our Principal, **Dr. Challa Dhanunjaya Naidu,** and our Head of Department, **Dr. M.Raja Sekar**, for extending their cooperation in doing this project within the stipulated time.

We extend our heartfelt thanks to our guide, **Ashalatha.G,** and the project coordinators **Mr. P. Veeranjaneyulu and Dr. T. Preethi** for their enthusiastic guidance throughout the course of our project.

Last but not least, our appreciable obligation also goes to all the staff members of the Computer Science & Engineering department and to our fellow classmates who directly or indirectly helped us.

Mr. B. Maneesh Kumar (21071A6707)

Miss. K. Sai Praneetha (21071A6726)

Miss. K. Sai Sahithi (21071A6727)

Mr. T. Sathwik (21071A6753)

ABSTRACT

The YouTube Transcript Summarizer is an innovative tool designed to streamline information retrieval from video content on the platform. Leveraging advanced natural language processing and machine learning algorithms, this tool meticulously analyzes the transcript of a YouTube video, identifying crucial keywords, key phrases, and prominent themes. By distilling the content into a succinct summary, users can quickly grasp the main points and essential information without having to invest time in watching the entire video. This feature not only enhances accessibility for users with time constraints but also facilitates efficient content consumption, making it a valuable resource for individuals seeking concise insights from a diverse range of video content on YouTube.

Through its intelligent abstraction of information, the YouTube Transcript Summarizer contributes to a more user-friendly and time-efficient experience on the platform. Users can now make informed decisions about which videos to explore further, based on the summarized content provided by the tool. Additionally, this summarization technology proves beneficial for content creators, as it enables them to reach a broader audience by offering a quick preview of their video's content. As a result, the YouTube Transcript Summarizer represents a significant advancement in optimizing the consumption and dissemination of information within the expansive and dynamic landscape of YouTube video content.

INDEX

List of Tables

1. INTRODUCTION

The YouTube Transcript Summarizer represents a groundbreaking advancement in content consumption on the popular video-sharing platform. As the sheer volume of videos on YouTube continues to grow exponentially, navigating through this vast sea of content can be time-consuming and overwhelming. In response to this challenge, the Transcript Summarizer leverages cutting-edge technologies, including natural language processing and machine learning, to extract meaningful insights from video transcripts. By condensing the information into succinct summaries, the tool provides users with a quick and efficient way to understand the key points of a video, making it an invaluable resource for both casual viewers and researchers seeking specific information.

One of the key strengths of the YouTube Transcript Summarizer lies in its ability to democratize information access. Users with limited time or those looking to quickly evaluate the relevance of a video to their interests can now rely on the summarizer to provide concise overviews. This not only enhances user experience but also promotes inclusivity by catering to a diverse audience with varying time constraints and preferences. Content creators, in turn, benefit from a tool that can potentially increase the visibility of their content by making it more accessible to a broader audience through summarized previews.

As the digital landscape continues to evolve, the YouTube Transcript Summarizer stands out as a pivotal tool, addressing the evolving needs of users and content creators alike. By bridging the gap between the vastness of video content and the time constraints of users, this tool exemplifies the transformative potential of technology in shaping the way we engage with online multimedia.

2. LITERATURE SURVEY/ EXISTING SYSTEM

2.1 FEASIBILITY STUDY

The YouTube Transcript Summarizer's feasibility lies in its technical capacity to develop robust NLP algorithms and seamlessly integrate with YouTube's API. Market demand, revenue models, legal compliance, and ethical considerations are crucial factors influencing its success. Additionally, operational aspects such as usability and ongoing maintenance contribute to determining the tool's viability.

2.1.1 ORGANIZATIONAL FEASIBILITY

The organizational feasibility of a YouTube Transcript Summarizer involves assessing the organization's resources, including skilled personnel and technological infrastructure. It requires evaluating the organizational culture for adaptability to technological advancements and ensuring strategic alignment with broader goals.

2.1.2 ECONOMIC FEASIBILITY

The economic feasibility of a YouTube Transcript Summarizer involves analyzing the potential financial impact and return on investment for the organization. This assessment includes estimating development costs, identifying revenue streams, and evaluating the tool's market potential.

2.1.3 TECHNICAL FEASIBILITY

The technical feasibility of a YouTube Transcript Summarizer centers on its ability to develop sophisticated natural language processing (NLP) algorithms and seamlessly integrate with YouTube's API. Evaluating scalability for handling diverse video content and ensuring compatibility with existing technology infrastructure are crucial considerations.

2.1.4 BEHAVIORAL FEASIBILITY

Behavioral feasibility for a YouTube Transcript Summarizer involves assessing user acceptance and adaptability to the tool. Understanding how users perceive and interact with summarized content is crucial. The tool's success depends on addressing user preferences, ensuring ease of use, and aligning with the behavior patterns of YouTube's diverse audience.

2.2 LITERATURE REVIEW

Ilampiray et al. [1]: The website can save time for the user .Instaed of Seeing the whole buffer waste content of the video it will prefix see the what is main content of YouTube Video and Know which video will perfectly for us. It save the time and effort of the user. By using the website their burden will be reduced for search the right YouTube Video. The website will also provide Multi-Language Summarization and made the availability to Text-toSpeech Process also. The approach aims to provide users with only the relevant and useful information on the topics that interest them, eliminating the need to watch lengthy videos. This time saved can be utilized for further knowledge acquisition and exploration.

Reshma Shaik et al. [2]: In this project they provided one platform for all summarizers such as article, file, URL and YouTube video summarizer. With help of Python they have made a graphical user interface (GUI) which provides one platform for all these summarizers and also they have used genism module which is a powerful tool for working with large collections of text data and performing unsupervised machine learning tasks such as topic modeling and document similarity analysis. It is widely used in both research and industry applications.Also,with the help of Spacy and nltk we have made our summarizer more accurate and easierto use. They have also used transformers which is also an open-source library for Natural Language Processing (NLP) developed by Hugging Face. The title of the summary can also be used to describe the text, which will boost the accuracy of the summary. The title of the summary describes the text, such as what the material is about.Extractive summarization involves selecting sentences fromthe source document to create a summary.

Rand Abdulwahid et al. [3]: The summaries that create automatically may not be as cohesive and smart as the summarization created by humans. The readers, on the other hand, were able to comprehend the produced summaries for the most part. Furthermore, the large number of text documents and the long video on the web introduces to the user summary of every document with more facility to find the desired documents or the suitable video. The automatic summary feature allows the user to quickly obtain a sense of all the material in the document and to identify the papers that are most relevant to the user's requirements. In this research, They shown the automatic summary for the transcription of YouTube long videos. Firstly, they fetched the text from the video, execute the pre-processing steps in the English text, and expound the text summarization process based on extractive type. Then, the algorithm of TF-IDF was used to summarize the transcript of the YouTube video depending on the important sentence in it. Through this experiment, it can be seen that it is a suitable method is used for extractive summary. By using this method, TF-IDF had proven that it is a strong method to produce the value that decides which word inside the text is important. That value assists the program for choose which sentence can be used in the summary. The dataset that used in this study was CNN-dailymail-master dataset.

Yogendra Singh et al. [4]: This work provides insight into the development of text-based video summarization. They scrutinize various methods in this review and offer a detailed comparison to support ongoing research in this domain. The critical insight derived from this study is the superior performance of supervised learning methods compared to unsupervised and weakly supervised techniques, which is contingent on the training effectiveness and dataset size. This modest contribution aims to aid individuals in selecting an appropriate technique for personalized video summarization.

Bhagat et al. [5]: This project proposes solution to summarize transcripts of lengthy youtube videos. In this project they have used several algorithms to effectively summarize the transcript. It gives users six different algorithms to choose and summarize the transcript. This application first retrieves the subtitles/transcript provided by the uploader or the automatic generated subtitles/transcript of the video from youtube_transcript_api available in python using the video link entered by the user. The application takes video link, summary algorithm and summary percentage as input and provides the summary of the transcript as output. This solution works only if the transcript is provided by the author or auto-generation is allowed while uploading the video, the project can be expanded by improving the application such that it work for the videos without transcript or auto-generation allowed.

Inamdar et al. [6]: A YouTube Transcript summarizer has been proposed for this project. The system uses the YouTube video the user has chosen as input and the Python API to acquire the video's transcripts when a user clicks the summarise button on the webpage for the Chrome extension. The transformers package is then used to summarise the transcripts that have been accessed. The user is then shown the summarised text in the chrome extension web page. This project greatly benefits users by saving them valuable time and resources.

2.3 EXISTING SYSTEM

- **Extraction Techniques:** Techniques for extracting key phrases or keywords from the transcript are commonly employed. These methods help in identifying the most significant elements of the video content.
- **Machine Learning Models:** Machine learning models, including supervised and unsupervised learning approaches, may be used to train algorithms for summarization. These models learn from large datasets to improve the accuracy of content condensation.
- **Semantic Analysis:** Semantic analysis plays a crucial role in understanding the context and meaning of the content. Existing systems might incorporate semantic analysis to ensure that the summarized content retains its original intent.

2.4 DRAWBACKS OF THE EXISTING SYSTEM

- Scalability Issues.
- Challenge in Handling User-Generated Content.
- Limited User Customization.

3. SOFTWARE REQUIREMENT ANALYSIS

3.1 INTRODUCTION

The software requirement analysis for a YouTube Transcript Summarizer involves identifying and documenting the essential features, performance criteria, and integration specifications to guide the development of an efficient and user-friendly summarization tool.

DOCUMENT PURPOSE

The goal is to Efficient YouTube video content summarization.

3.1.1 DEFINITIONS

Machine learning
Machine learning in a YouTube Transcript Summarizer encompasses training models with supervised learning, using sequence-to-sequence and attention mechanisms, employing transfer learning for optimization, and integrating continuous learning. Evaluation metrics like ROUGE guide the refinement process for improved summarization accuracy and adaptability to diverse video content.

Data preprocessing
Data preprocessing in a YouTube Transcript Summarizer, especially when incorporating Natural Language Processing (NLP) models, involves several essential steps to prepare the textual data for effective analysis and summarization. Here's an overview of key data preprocessing steps:

- Text Cleaning
- Tokenization
- Lowercasing
- Stopword Removal
- Lemmatization or Stemming
- Handling Special Characters and Numbers
- Handling Missing Data
- Handling HTML Tags and URLs
- Sentence Segmentation
- Encoding Text

One Hot Encoding:
One-hot encoding in a YouTube Transcript Summarizer involves representing words from video transcripts as binary vectors, where each word is uniquely encoded. This technique creates a sparse matrix, simplifying the input data for machine learning models. One-hot

encoding facilitates efficient processing of textual information, enabling summarization algorithms to work with discrete representations of words in the context of YouTube content analysis.

Integration Specifications:

The detailed requirements and guidelines for how a software system, such as a YouTube Transcript Summarizer, should seamlessly integrate with external components or APIs, ensuring interoperability and compatibility with other platforms or services.

K-Fold Cross Validation:

K-Fold Cross Validation in a YouTube Transcript Summarizer is a resampling technique where the video transcript dataset is divided into K subsets, and the summarization model is trained and evaluated K times, using different subsets for training and validation in each iteration, providing a robust estimate of its generalization performance.

1. Split the YouTube video transcript dataset into K subsets.
2. Train the summarization model K times, utilizing K-1 subsets for training and 1 subset for validation in each iteration.
3. Evaluate the summarizer's performance for each iteration, considering metrics like summary coherence and informativeness.
4. Calculate the average performance metric across all iterations to assess the overall effectiveness of the YouTube Transcript Summarizer.

Ordinal Data:

Ordinal data in a YouTube Transcript Summarizer refers to information with a clear order but lacks consistent numerical intervals, such as sentiment scores (positive, neutral, negative). Integrating ordinal data enhances the model's contextual understanding for more nuanced summarization.

Nominal Data:

Nominal data in a YouTube Transcript Summarizer represents categorical information without a meaningful order, such as video genres (science, entertainment). It is useful for categorization in the summarization process, enhancing organization and customization of summaries based on non-hierarchical attributes.

3.2 SYSTEM ARCHITECTURE

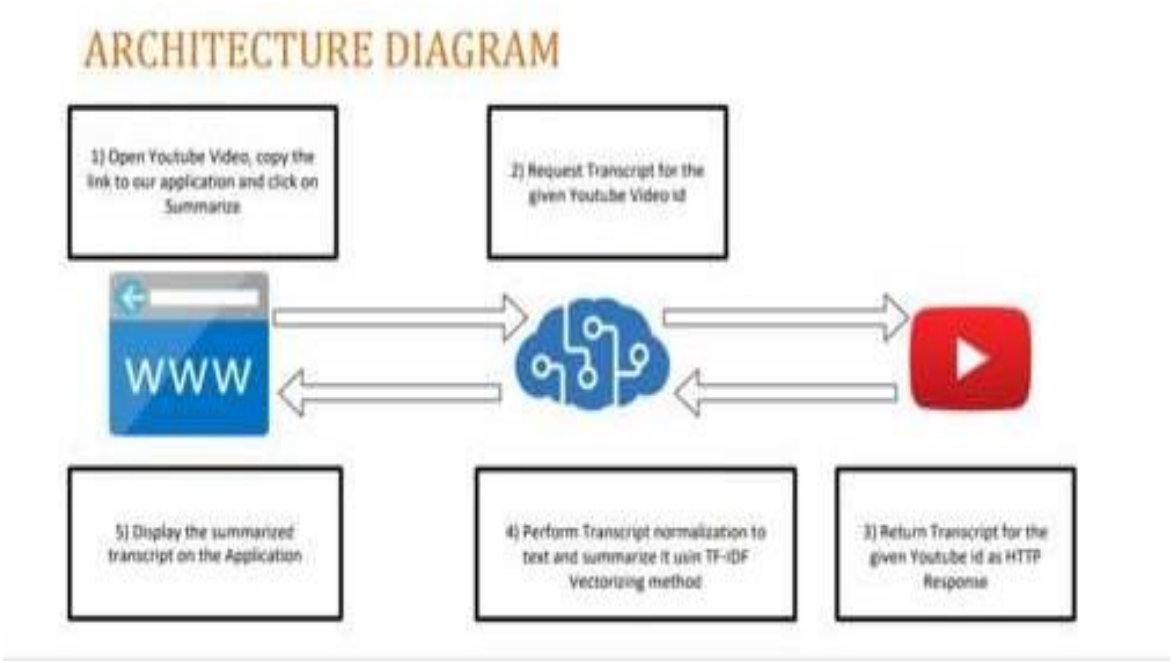


Fig 3.2.1: System Architecture

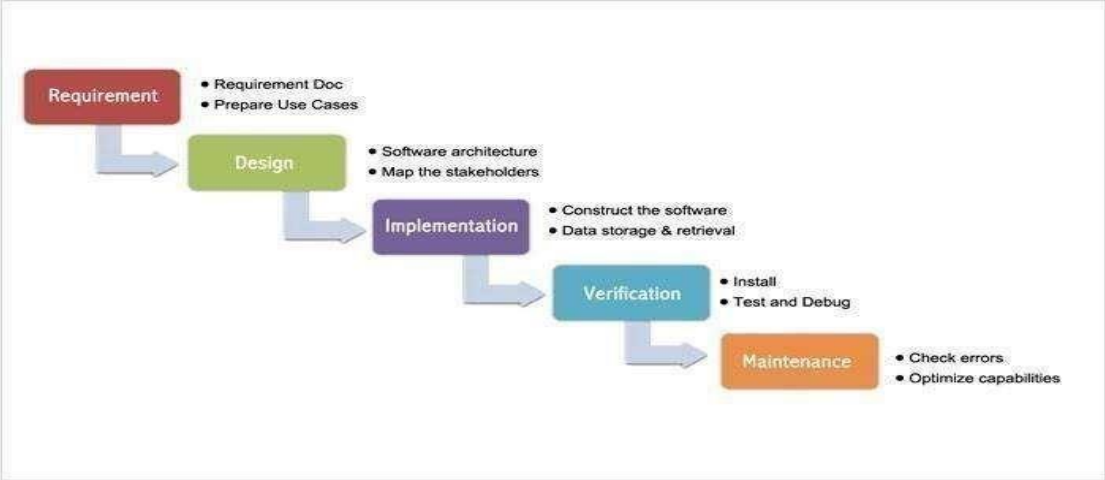
The architecture of a YouTube transcript summarizer with NLP involves utilizing the YouTube API for data retrieval, employing Automatic Speech Recognition or leveraging existing transcripts, preprocessing the text through cleaning, tokenization, and entity recognition, followed by sentiment analysis. The summarization process encompasses both abstractive and extractive techniques, extracting keywords and organizing content coherently. A user-friendly interface, potentially integrated with YouTube, facilitates user interaction, while a feedback system aids model improvement. Scalability, performance optimization, and security measures ensure the system's efficiency and compliance with privacy standards. Continuous model training and adaptation to user feedback contribute to ongoing enhancements in summarization accuracy and language pattern recognition.

The architecture of a YouTube Transcript Summarizer typically includes components for data collection, preprocessing, feature extraction, and a model (e.g., sequence-to-sequence or transformer) with attention mechanisms. Training involves using labeled datasets, and evaluation metrics like ROUGE are employed. Continuous learning, hyperparameter tuning, and optional user interfaces contribute to an effective summarization system.

3.3 FUNCTIONAL REQUIREMENTS

- Transcript Retrieval
- Preprocessing
- Summarization Model

3.4 SYSTEM ANALYSIS



It was the first Process Model to be introduced. A linear sequential life cycle model is another name for it. It's quite simple to use and understand. Phases do not overlap in this paradigm, and each phase must be finished before the next one begins. The first SDLC approach used during software development was the waterfall model.

The model shows that the development of software is linear and is a sequential process. Only after one phase of the development is completed, we can go to the next phase. In this waterfall paradigm, the phases do not overlap.

The steps in the waterfall model are explained below.

Requirements: The search has become more intense and concentrated on the software's requirements at this time. To comprehend the nature of the programs to be developed, the software engineer must first comprehend the software's information domain, which includes the required functionalities, user interface, and so on. The customer must be informed about the second activity, which must be recorded and presented.

Design: This step is used to transform the above criteria as a representation in the form of "blueprint" software before coding begins. The design must be able to meet the criteria laid out in the previous stage.

Implementation: The design was converted into a machine-readable format in order for it to be interpreted by a computer in some circumstances, i.e., through the coding process into a programming language. This was the stage in which the programmer will put the technical design phase into action.

Verification: It, like anything else constructed, must first be put to the test. The same may be said for software. To ensure that the application is error-free, all functions must be checked, and the results must closely comply to the previously specified requirements.

Maintenance: Software maintenance, including development, is essential since the software that is being generated is not always exactly like that. It may still have minor faults that were not identified previously when it runs, or it may require additional capabilities that were not previously available in the software.

Useful factors: The Waterfall model has its advantages like it is simple to use. Additionally, while using the model all the system requirements can be defined as a whole, explicitly and at the start the product can run without many issues.

It is economic to make changes in the early stages of the project when there are problems with system requirements then when the problems which arise in later stages.

3.5 NON-FUNCTIONAL REQUIREMENTS

Ease of Use

- The system is simple, user friendly.

Extensibility

- The system can be easily extended to incorporate additional functionality.

Security

- The system is secure because it doesn't share any information of the client without his/her consent.

Maintainability

- The system will be as self-contained as possible to allow for ongoing maintenance.

Reliability

- Most efficient model is chosen that is linear regression or XGBOOST with 80% accuracy

3.6 SOFTWARE REQUIREMENT SPECIFICATION

JUPYTER NOTEBOOK

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use

GOOGLE COLAB

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook

- Write and execute code in Python
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

3.7 SOFTWARE REQUIREMENTS

- Software : Jupyter notebook or google colab
- Operating System : Windows family
- Technology : Machine Learning

3.8 HARDWARE REQUIREMENTS

- Minimum 8GB Ram Laptop
- Internet Connection

4. SOFTWARE DESIGN

4.1 UML DIAGRAMS

The Device Architecture Manual describes the application requirements, operating state, application and subsystem functionality, documents and repository setup, input locations, yield types, human-machine interfaces, management reasoning, and external interfaces. The Unified Modeling Language (UML) assists software developers in expressing an analysis model through documents that contain a plethora of syntactic and semantic instructions. A UML context is defined as five distinct viewpoints that present the system from a particularly different point of view.

The components are similar to modules that can be combined in a variety of ways to create a complete UML diagram. As a result, comprehension of the various diagrams is essential for utilizing the knowledge in real-world systems. The best method to understand any complex system is to draw diagrams or images of it. These designs have a bigger influence on our understanding. Looking around, we can see that info-graphics are not a new concept, but they are frequently utilized in a variety of businesses in various ways.

User Model View

The perspective refers to the system from the clients' point of view. The exam's depiction depicts a situation of utilization from the perspective of end-clients. The user view provides a window into the system from the perspective of the user, with the system's operation defined in light of the user and what the user wants from it.

Structural model view

This layout represents the details and functionality of the device. This software design maps out the static structures. This view includes activity diagrams, sequence diagrams and state machine diagrams

Behavioral Model View

It refers to the social dynamics as framework components, delineating the assortment cooperation between various auxiliary components depicted in the client model and basic model view. UML Behavioral Diagrams illustrate time-dependent aspects of a system and communicate the system's dynamics and how they interact. Behavioral diagrams include interaction diagrams, use case diagrams, activity diagrams and state–chart diagrams.

Implementation Model View

The essential and actions as frame pieces are discussed in this when they are to be

manufactured. This is also referred to as the implementation view. It uses the UML Component diagram to describe system components. One of the UML diagrams used to illustrate the development view is the Package diagram.

Environmental Model View

The systemic and functional component of the world where the program is to be introduced was expressed within this. The diagram in the environmental view explains the software model's after-deployment behavior. This diagram typically explains user interactions and the effects of software on the system. The following diagrams are included in the environmental model: Diagram of deployment.

The UML model is made up of two separate domains:

- Demonstration of UML Analysis, with a focus on the client model and auxiliary model perspectives on the framework.
- UML configuration presenting, which focuses on demonstrations, usage, and natural model perspectives.

4.1.1 USE CASE DIAGRAM

The objective of a use case diagram is to portray the dynamic nature of a system. However, because the aim of the other four pictures is the same, this description is too broad to characterize the purpose. We'll look into a specific purpose that distinguishes it from the other four diagrams.

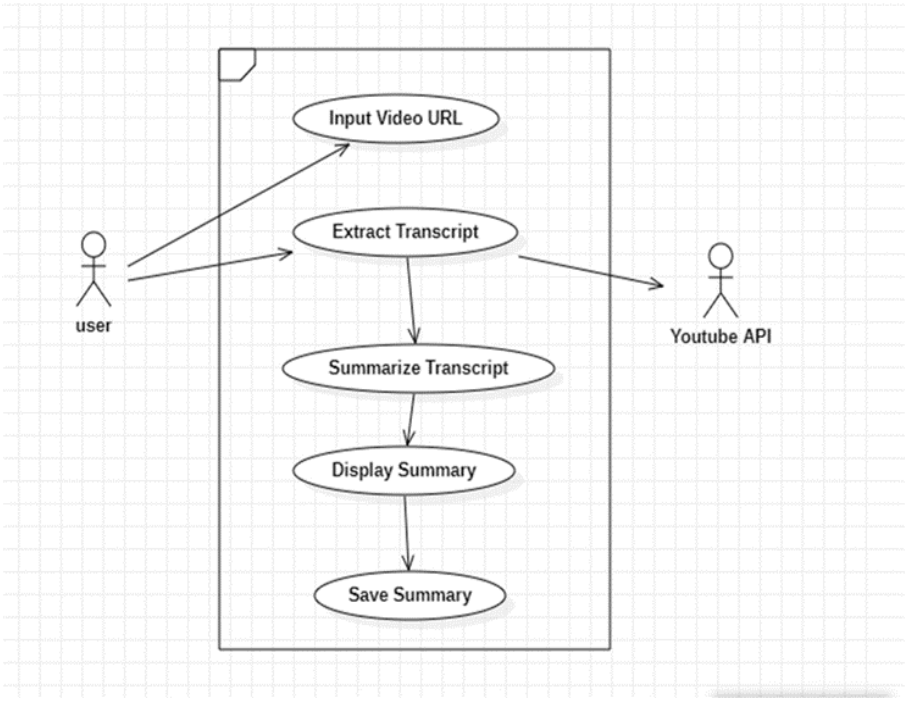


Fig 4.1.1.1: Use Case diagram for the application

Actors:

- User
- Youtube API

Use Cases:

- Input data URL
- Extract transcript
- Summarise transcript
- Display summary
- Save summary

Connections:

- User to Use Cases
- YouTube API to Extract Transcript
- Summarize Transcript uses Extracted Transcript
- Display Summary uses Summarized Transcript
- Save Summary uses Summarized Transcript

4.1.2 SEQUENCE DIAGRAM

Because it illustrates how a group of items interact with one another, a sequence diagram is a form of interaction diagram. These diagrams are used by software engineers and business people to comprehend the requirements for a new system or to document a current process. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams can be useful as a reference for businesses and other organizations. Make the diagram to show:

- Describe the specifics of a UML use case.
- Create a model of the logic of a complex procedure, function, or operation.
- Examine how objects and components interact with one another in order to complete a process.
- Plan and comprehend the specific functionality of a current or future scenario.

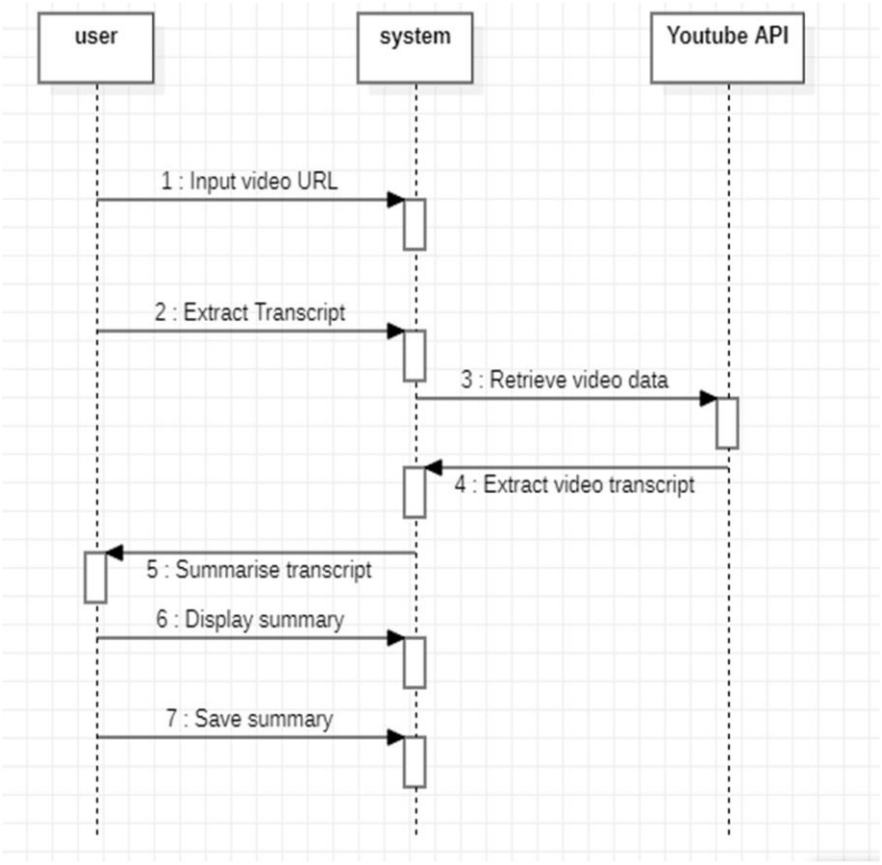


Fig 4.1.2.1: Sequence Diagram for 3D model display

In the above sequence diagram, the lifelines are:

- User
- System
- Youtube API

The sequence starts from the user inputs the video URL. The system initiates the process by extracting the video transcript. The system communicates with the YouTube API to retrieve video data. The YouTube API extracts the video transcript. The system processes the transcript to generate a summary. The system displays the summary to the user. The user has the option to save the summary. The system saves the summarized data.

4.1.2 ACTIVITY DIAGRAM

An activity diagram is a flowchart that displays the movement of information from one action to the next. A system operation can be used to describe the activity.

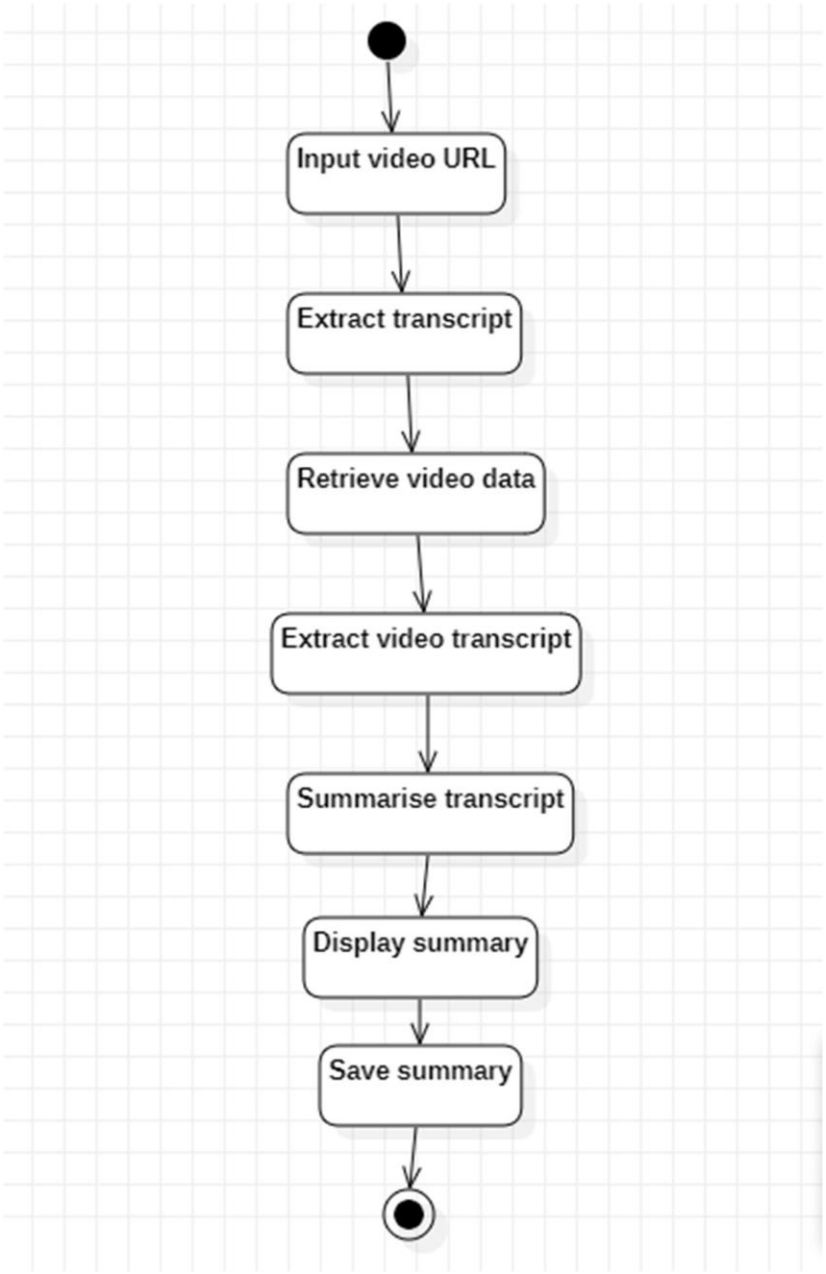


Fig 4.1.3.1: Activity Diagram

This activity diagram shows the whole activity of the system. The Activity starts with the user inputs the video URL. The system extracts the video transcript. The system communicates with the YouTube API to retrieve video data. The YouTube API extracts the video transcript. The system processes the transcript to generate a summary. The system displays the summary to the user. The user has the option to save the summary. The system saves the summarized data.

4.1.3 CLASS DIAGRAM

A static diagram is also referred to as a class diagram. It depicts the static view of an application. A class diagram can be used to visualize, describe, and document various parts of a system, as well as to create executable code for a software programmer. The traits and activities of a class, as well as the constraints, are described in a class diagram.

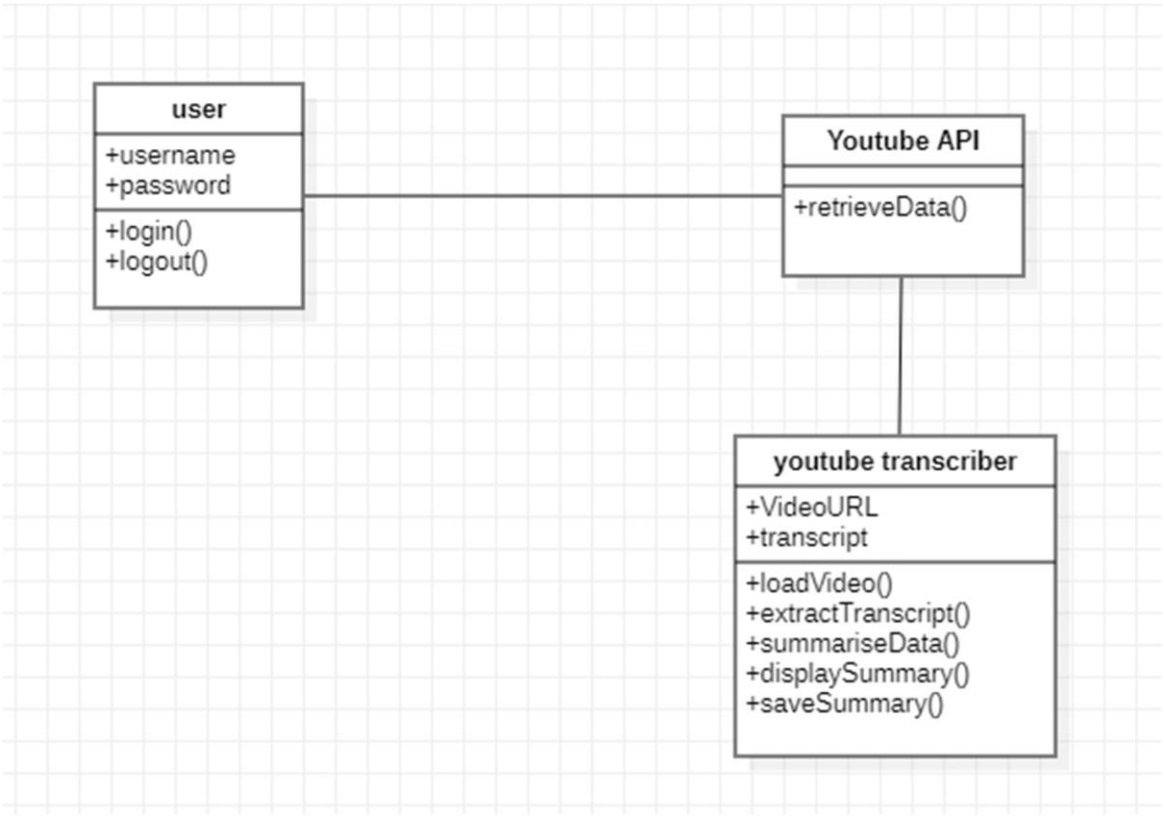


Fig 4.1.4.1: Class Diagram

User Class: Represents the user of the system.Attributes: username and password.

Operations: login (), logout().

YouTube API Class: Represents the interface with YouTube's data.Operations: retrieve Data().

Transcript Class: Represents the transcript of a video.

Attributes: content and language.

Operations: getText(), getLanguage().

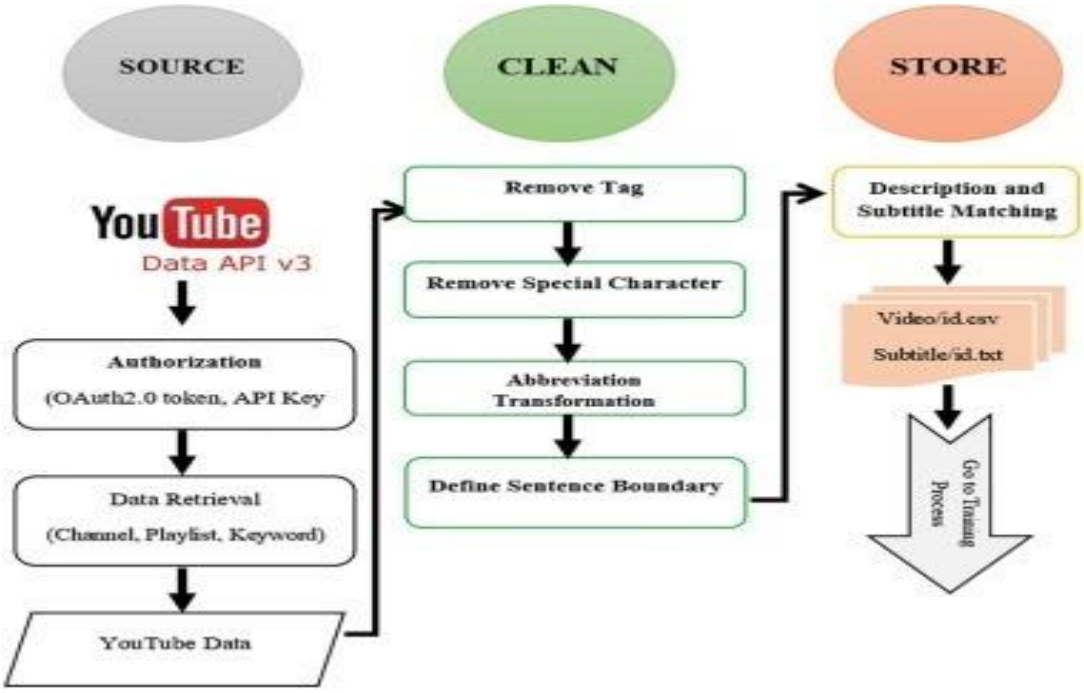
YouTubeTranscriber Class: Represents the main class for the YouTube transcript summarizer system.

Attributes: videoURL and transcript.

Operations: loadVideo(), extractTranscript(), summarizeTranscript(), displaySummary(), saveSummary(),provideFeedback().

5 PROPOSED SYSTEM

5.1 METHODOLOGY



Data Set:

Data is crucial in finding pattern and use that pattern to predict the outcome. Collect YouTube videos for which we can obtain transcripts. We may use the YouTube API to retrieve video information and transcripts.

Cleaning:

Text Preprocessing:

- Convert the transcript to lowercase.
- Remove any unnecessary symbols, emojis, or special characters.
- Remove URLs or hyperlinks.
- Handle contractions and possessives.

Speaker Identification:

- If the transcript includes speaker names, consider identifying and removing them.
- You may use regular expressions or Named Entity Recognition (NER) for this task.

Remove Timestamps:

- YouTube transcripts often contain timestamps. Remove them, as they might not be relevant to summarization.

Handle Inaudible or Unintelligible Speech:

- Replace or remove sections marked as "inaudible" or "unintelligible."

Remove Repetitions:

- Detect and remove repeated phrases or sentences.

Filter Out Noise:

- Identify and filter out non-speech elements, like applause, background noise, or music, if present.

Models:

A To create a YouTube transcript summarizer, use the YouTube API for transcript retrieval, apply models like BERTSUM or GPT-3 for extractive or abstractive summarization, respectively. Employ regular expressions and Named Entity Recognition for data cleaning, and fine-tune models on domain-specific data if needed. Evaluate the summarizer using metrics like ROUGE scores and integrate it into the YouTube API or a user interface for practical use.

In Our Project we compared different machine learning models like

- BART (BART: Denoising Sequence-to-Sequence Pre-training)
- BERT (Bidirectional Encoder Representations from Transformers)
- GPT (Generative Pre-trained Transformer) Models

5.2 FUNCTIONALITIES

5.2.1 Importing Data:

- we import You tube videos
- link contains video that we want to summarize

5.2.2 Preprocessing data:

- Removing inconsistencies in data like ordering the links
- Checking if link is valid or not

5.2.3 Choosing Model:

- Test each model and choose best model according to accuracies

5.2.4 Summarizing data:

- Data is summarized and displayed
- We can view the saved data

5.3 ADVANTAGES OF PROPOSED SYSTEM:

- Efficient Content Consumption
- Time-Saving
- Improved Accessibility
- Enhanced Searchability
- Content Creators' Insights

6.CODING AND IMPLEMENTATIONS

6.1 DATASET:

It is used to train the model. The model learns from this data set and gives the outputs based on thelearning during the testing. Our dataset is you tube which contains zillions of videos with closed captions that can be summarized.

6.2 Understanding Data:

Initially we imported all libraries which are required. Data is in the video format. This data is givenas the input to the code and the code generates the subtitles for it then after we summarize the data.

```
[ ]  !pip install -q transformers

[ ]  !pip install -q youtube_transcript_api

[ ]  from transformers import pipeline
     from youtube_transcript_api import YouTubeTranscriptApi

[ ]  youtube_video = "https://www.youtube.com/watch?v=9RhWXPcKBI8"

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

[ ]  video_id

     '9RhWXPcKBI8'

[ ]  from IPython.display import YouTubeVideo
     YouTubeVideo(video_id)
```

```
[ ] from IPython.display import YouTubeVideo
    YouTubeVideo(video_id)
```


```
[ ] YouTubeTranscriptApi.get_transcript(video_id)
    transcript = YouTubeTranscriptApi.get_transcript(video_id)
```

Now we will find the length of the transcript for the video

```
summarizer = pipeline('summarization')

No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 and
Using a pipeline without specifying a model name and revision in prod
config.json: 100%  1.80k/1.80k [00:00<
pytorch_model.bin: 100%  1.22G/1.22G
tokenizer_config.json: 100%  26.0/26.0
vocab.json: 100%  899k/899k [00:00<0
merges.txt: 100%  456k/456k [00:00<0
```

```
[ ]  num_iters = int(len(result)/1000)
     summarized_text = []
     for i in range(0, num_iters + 1):
       start = 0
       start = i * 1000
       end = (i + 1) * 1000
       print("input text \n" + result[start:end])
       out = summarizer(result[start:end])
       out = out[0]
       out = out['summary_text']
       print("Summarized text\n"+out)
       summarized_text.append(out)

     print(summarized_text)
```

```
input text
 I've built a massive
isolation chamber and we're going to see
if these two strangers can survive in this cube
for the next 100 days. They have never met each other ever. Bailey. This is Suzie. - Suzie, this is Bailey.
- Nice to meet you. Hi! Nice to meet you. If the two of you can survive
the next 100 days in here. I will give you the half a million
dollars inside of this vault. But if one of you leaves before the 100
days is up, you both get nothing. All right. I think you
guys understand the rules. - Have fun.
- Okay. Bye. This is going to be crazy. Yeah. This is actually like an insane
asylum. They're currently looking at all
the stuff we put in there. We gave them enough food
for 100 days, which is healthy,
but basically the exact same thing over and over again. We also gave them their own private bathroom,
which comes with the shower and obviously has no cameras inside
and a bed to sleep on. They have everything
they need to survive 100 days. It's just a question of do
they want
```

```
Summarized text
 Two strangers have never met each other ever in a massive isolation chamber . If they can survive the next 100
input text
it? I got a comb. I don't know if
I've ever combed my hair in my life. Interesting. See how they're both standing
on different sides of the room? They're so awkward. When I had them
take their blindfolds off. That was legitimately
the first time they had ever met. I just keep thinking of the feeling
like of exiting. Oh, yeah? How good is that going to feel? It's only a 100 sleeps. Only 100 sleeps. And then th
kept going. That's like, what freaks me out. The most interesting
part of this experiment is that in this room, you have no idea
what day or time it is. And already night one. Suzie and Bailey made the mistake
of only sleeping for 2 hours, which means they started
their second day at 2:37 a.m. I'm just going to just act like
this is just the new normal. - Okay, what do you want to eat?
- Are you feeling beef or chicken? - Chicken.
- Chicken sounds great. That is sickening. It's good. As you can see, living inside
a white cube is pretty boring. Besides eating
Summarized text
 Suzie and Bailey started their experiment at 2:37 a.m. on their second day after only sleeping for 2 hours . S
input text
the same thing over and over, time in the chamber
consisted of two things trying to hide
from the bright light. The lights are just so bright,
that we have to make these things to shield ourselves.  And desperate attempts to keep themselves entertained.
when they use the paper from their journals
to make a deck of playing cards. Is he making... Cards? All right.
```

Now we need to print the summerized text from the actual transcript.

```
len(str(summarized_text))
```

```
8407
```

```
str(summarized_text)
```

```
'[\' Two strangers have never met each other ever in a massive isolation chamber . If th
is actually like an insane psychiatric asylum. They\\\'re currently looking at all the s
their experiment at 2:37 a.m. on their second day after only sleeping for 2 hours . Suzi
It\\\'s only a 100 sleeps. Only 100 sleeps"\', \' Time in the chamber consisted of two t
ese things to shield themselves. And desperate attempts to keep themselves entertained.
s.\', " It\'s crazy to me that already on day 9 of this challenge, they\'re already bor
```

6.3 Data Preprocessing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model.It is the first and crucial step while creating a machine learning model.When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

Some preprocessing tasks we are going to do are:

1.Checking whether the source of the data is transparent or not?

2.Check the URL

6.4 Choosing model

Models:

A machine learning model is defined as a mathematical representation of the output of the training process. Machine learning is the study of different algorithms that can improve automatically through experience & old dataand build the model. A machine learning model is similar to computer software designed to recognize patterns or behaviors based on previous experience or data. The learning algorithm discovers patterns within the training data,and it outputs an ML model which captures these patterns and makes predictions on new data.

The key features of GPT (Generative Pre-trained Transformer) models include:

Transformer Architecture: GPT models are built on the Transformer architecture, which allows for efficient processing of sequential data through self-attention mechanisms.

Pre-training: GPT models are pre-trained on a large corpus of diverse text data. This pre-training helps the model learn language patterns, grammar, and semantic understanding.

Generative Nature: GPT models are generative, meaning they can generate coherent and contextually relevant text based on a given prompt. They are capable of completing sentences, paragraphs, or even generating longer pieces of text.

Unsupervised Learning: GPT is trained in an unsupervised manner, meaning it doesn't require labeled data for specific tasks. The model learns from the raw text input without explicit guidance on the desired output.

Autoregressive Decoding: GPT models use autoregressive decoding during text generation. This means the model generates one token at a time, considering the context of previously generated tokens.

7.TESTING

In machine learning, testing is mainly used to validate raw data and check the ML model's performance. The main objectives of testing machine learning models are:

- Quality Assurance
- Detect bugs and flaws

Once your machine learning model is built (with your training data), you need unseen data test your model. This data is called testing data, and you can use it to evaluate the performance and progress of your algorithms' training and adjust or optimize it for improved results.

Testing data has two main criteria. It should:

- Represent the actual dataset
- Be large enough to generate meaningful predictions

7.1 TYPES OF TESTING

7.1.1 MANUAL TESTING

Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application. Manual software testing is the most primitive technique of all testing types and it helps to find critical bugs in the software application.

Any new application must be manually tested before its testing can be automated. Manual Software Testing requires more effort but is necessary to check automation feasibility. Manual Testing concepts does not require knowledge of any testing tool. One of the Software Testing Fundamental is "**100% Automation is not possible**". This makes Manual Testing imperative.

7.1.2 AUTOMATED TESTING

Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.

The automation testing software can also enter test data into the System Under Test, compare

expected and actual results and generate detailed test reports. Software Test Automation demands considerable investments of money and resources.

7.2 TESTING LEVELS

7.2.1 NON-FUNCTIONAL TESTING

Non-functional testing is a type of software testing to test non-functional parameters such as reliability, load test, performance and accountability of the software. The primary purpose of non-functional testing is to test the reading speed of the software system as per non-functional parameters. The parameters of non-functional testing are never tested before the functional testing. Non-functional testing is also very important as functional testing because it plays a crucial role in customer satisfaction.

7.2.1.1 PERFORMANCE TESTING

Performance testing is a form of software testing that focuses on how a system running the system performs under a particular load. This is not about finding software bugs or defects. Different performance testing types measure according to benchmarks and standards. Performance testing gives developers the diagnostic information they need to eliminate bottlenecks.

7.2.1.2 STRESS TESTING

Stress Testing is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn't crash under crunch situations. It even tests beyond normal operating points and evaluates how software works under extreme conditions.

7.2.1.3 SECURITY TESTING

Security Testing is a type of Software Testing that uncovers vulnerabilities of the system and determines that the data and resources of the system are protected from possible intruders. It ensures that the software system and application are free from any threats or risks that can cause a loss. Security testing of any system is focused on finding all possible loopholes and weaknesses of the system which might result in the loss of information or repute of the organization.

7.2.1.4 PORTABILITY TESTING

Portability Testing is one of Software Testing which is carried out to determine the degree of ease or difficulty to which a software application can be effectively and efficiently transferred from one hardware, software or environment to another one. The results of portability testing are measurements of how easily the software component or application will be integrated into the environment and then these results will be compared to the non-functional requirement of portability of the software system.

7.2.1.5 USABILITY TESTING

Usability Testing also known as User Experience (UX) Testing, is a testing method for measuring how easy and user-friendly a software application is. A small set of target end-users, use software application to expose usability defects. Usability testing mainly focuses on user's ease of using application, flexibility of application to handle controls and ability of application to meet its objectives. This testing is recommended during the initial design phase of SDLC, which gives more visibility on the expectations of the users.

7.2.2 FUNCTIONAL TESTING

It is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification. In functional testing, each function tested by giving the value, determining the output, and verifying the actual output with the expected value. Functional testing performed as black-box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application. Functional testing also called as black-box testing.

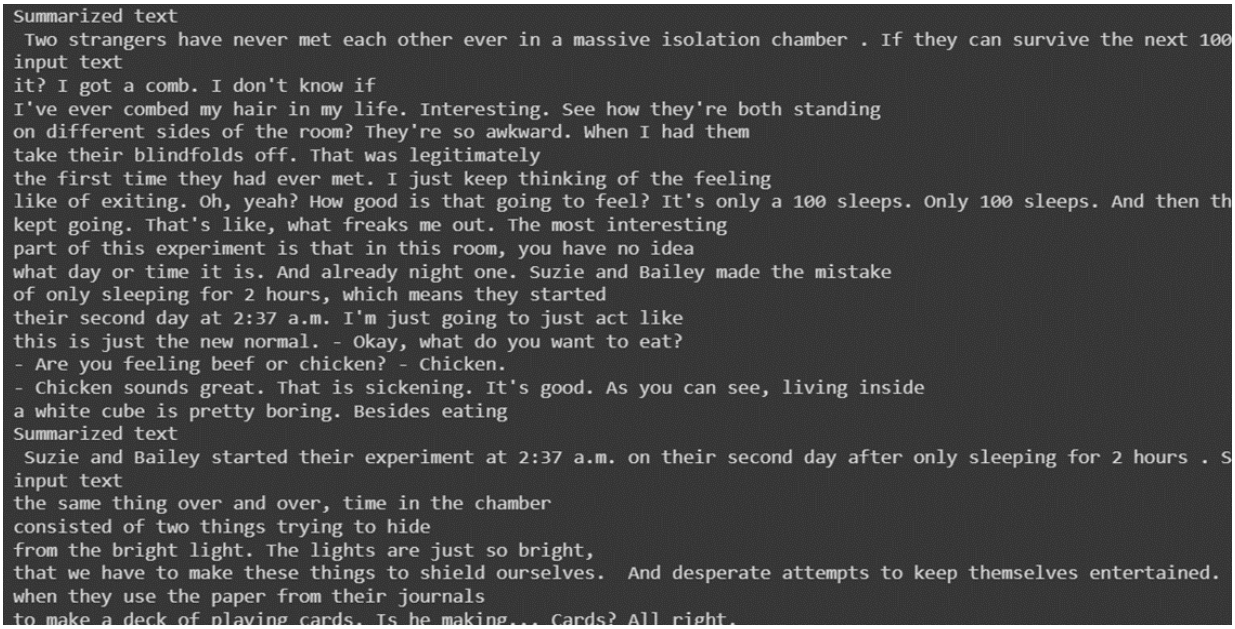
7.3 TEST CASES

Sl.no	Testcase	Expected Result	Actual Result	Pass/ Fail
1.	After Training testing with test sample data	most of the ou comes are righ upto 80 % accuracy	Summarized videos	PASS

8.RESULTS

```
str(summarized_text)
```

```
'[\' Two strangers have never met each other ever in a massive isolation chamber . If th
is actually like an insane psychiatric asylum. They\\\'re currently looking at all the s
their experiment at 2:37 a.m. on their second day after only sleeping for 2 hours . Suzi
It\\\'s only a 100 sleeps. Only 100 sleeps"\', \' Time in the chamber consisted of two t
ese things to shield themselves. And desperate attempts to keep themselves entertained.
s.\', " It\'s crazy to me that already on day 9 of this challenge, they\'re already bor
```



The YouTube transcript summarizer utilizes Natural Language Processing to generate concise summaries of video transcripts, distilling key information for users. The output includes a summarized text that captures the essential content of the video, providing users with a quick overview. Additionally, the system reports the length of the summarized text, allowing users to gauge the depth of the summary and decide on the level of detail they require. This feature enhances user control and customization, tailoring the summarization output to individual preferences and information needs.

9.CONCLUSION AND FURTHER WORK

The future scope of a YouTube transcript summarizer is likely to involve advancements in technology, user preferences, and content creation. The potential areas of future development andgrowth for YouTube transcript summarizers:

1. Improved Summarization Models:

   Advancements in natural language processing (NLP) and machine learning may lead to more sophisticated summarization models. These models could better capture context, generate more coherent summaries, and handle a broader range of content types.

2. Multimodal Summarization:

   Future summarizers might incorporate not only textual information from transcripts but also visual and auditory elements from the videos. This could lead to more comprehensive and informative summaries.

3. Customization and Personalization:

   Summarizers may become more personalized, allowing users to customize the level of detail

in summaries, select specific topics of interest, or adapt to individual learning preferences.

4. Real-time Summarization:

   As technology progresses, real-time summarization of live streams or rapidly changing content may become more feasible. This could be valuable for news coverage, live events, or educationalstreams.

5. Enhanced Accessibility Features:

   Future summarizers may focus on improving accessibility features, catering to diverse user needs, including those with disabilities. This could involve better support for multiple languages, improved readability, and enhanced compatibility with assistive technologies.

6. Integration with Learning Platforms:

   Integration with educational platforms and learning management systems could be a future trend. Summarizers might offer features that assist students in summarizing educational videos, generating study materials, and facilitating better knowledge retention.

REFERENCES

[1] Ilampiray, P., A. Thilagavathy, A. S. Nithin, and Infant Raj. "Video Transcript Summarizer." In E3S Web of Conferences, vol. 399, p. 04015. EDP Sciences, 2023.

[2] Reshma Shaik, Saloni Bargat, and Prof. Shilpa Ghode. 2023. “Article and YouTube Transcript Summarizer Using Spacy and NLTK Module”. SSGM Journal of Science and Engineering 1 (1):126-31. https://ssgmjournal.in/index.php/ssgm/article/view/46.

[3] Albeer, Rand Abdulwahid, Huda F. Al-Shahad, Hiba J. Aleqabie, and Noor D. Al-shakarchy. "Automatic summarization of YouTube video transcription text using term frequency-inverse document frequency." Indonesian Journal of Electrical Engineering and Computer Science 26, no. 3 (2022): 1512-1519.

[4] Singh, Yogendra, Rishu Kumar, Soumya Kabdal, and Prashant Upadhyay. "YouTube Video Summarizer using NLP: A Review." International Journal of Performability Engineering 19, no. 12 (2023): 817.

[5] Bhagat, Amit, and Prasanna Anjankar. "TEXT SUMMARIZATION ON YOUTUBE VIDEOS IN EDUCATIONAL DOMAIN."

[6] Inamdar, Eesha, Varada Kalaskar, and Vaidehi Zade. "SURVEY PAPER ON YOUTUBE TRANSCRIPT SUMMARIZER."