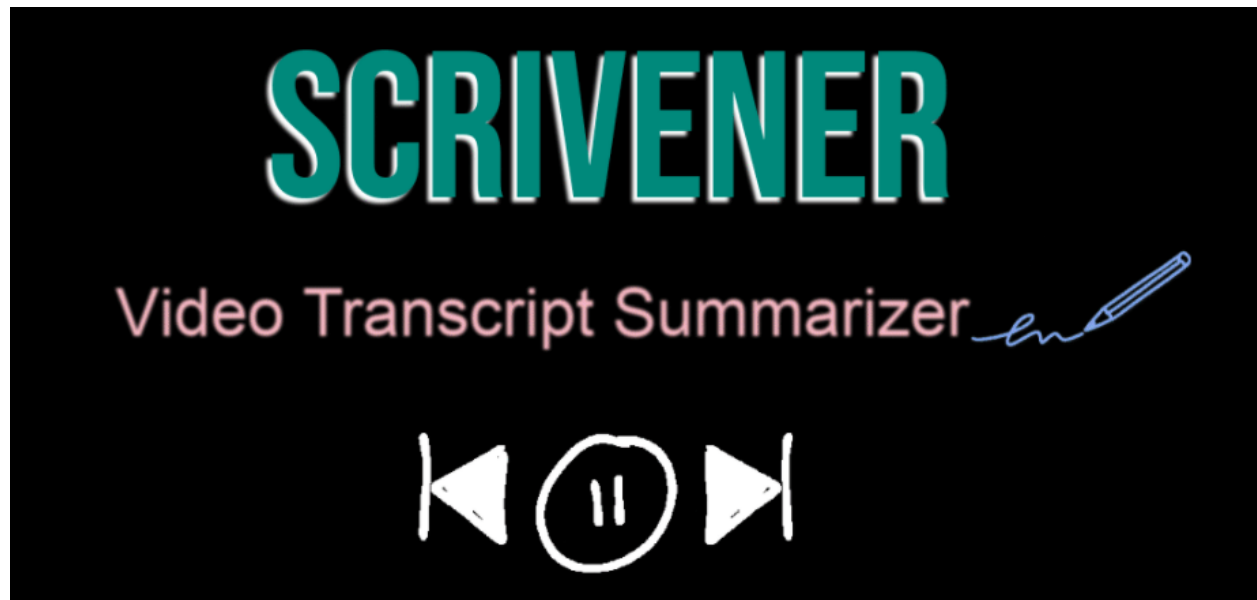


## Youtube Scrivener



### **Team Members:**

1. Anshul Navinbhai Patel
2. Bhavya Omprakash Agrawal
3. Darshan Manharbhai Patel
4. Pragna Bollam
5. Rohan Jigarbhai Shah

### **Introduction:**

Scrivener is a video transcript summarizer for Youtube videos. Youtube is one of the most used websites. A lot of people use the captions to understand the language of the video. In our project we aim to create a transcript summarizer which accepts a youtube URL link, collects the caption at every sentence and then provides the summary of the complete video. Our goal is to make the summarizer as accurate as possible and to add various other features. Our second goal of the project is to create a summarizer which can summarize the youtube videos which have captions disabled. Our project can be further expanded for numerous applications. This document provides a major perspective for the users to understand and take up the project as an Open source software and add on multiple features. Also, the document aids the developers in understanding the code and acts as a reference point for starting the project.

The complete development was achieved using the Python3 technology and it is recommended that the next set of developers who take up this project have these technologies installed and keep them running before proceeding further.

**Motivation:**

People watch long Youtube videos but their actual content can be summarized in small paragraph which takes 5 mins to read. And there are many videos which do not have closed captions (cc). For such videos, it would be helpful for people if there exists a software to convert the video audio into text and summarize it to save time. So, for these reasons, an automated summarizer would be helpful to people primarily to save time.

**Type of User who uses Scrivener:**

Inorder to save time on watching youtube videos, people can opt to use this to summarize the content. Other users can also upload their videos without cc to get a summary of the content.

**Third-party dependencies:**

1. **StreamLit:** It is an open-source Python library that allows users to create custom web applications.
2. **YouTubeTranscriptApi:** It is a Python Api that allows users to get transcripts/ subtitles from Youtube videos.
3. **Speech\_recognition:** Library to recognize speech and convert into text tokens.
4. **Moviepy:** A python module for video editing.
5. **Pydub:** Manipulates audio with an easy and high level interface.

**Steps for Execution:**

1. Clone the Git repository.
2. Run `pip install -r requirements.txt` followed by `python -c`
3. Open Command Prompt and change the directory to the location of the cloned repository.
4. Run the command `"streamlit run user_interface.py"`
5. Next, open your browser and type in `localhost:8501` in the search bar to open the webUI of the application.
6. The UI typically looks as shown below and here you have a choice between URL, file or normal text input.

## Code Functionalities:

This section contains details of the major source code files implemented in this project.

### 1. `transcribe.py` -

#### a. Class: `TranscribeVideo`

##### i. Function: `transcribe_video`

**Description:** Generate summary from youtube video without closed captions or uploaded video

**Inputs:**

- `ip_path`: input path where the uploaded video is stored temporarily

**Output:** returns the summary generated for that video in text format

##### ii. Function: `split_init`

**Description:** Split a large audio file into multiple smaller chunks of 2 mins each and store them in a temporary location.

**Inputs:** None

**Output:** returns number of smaller audio files created

### 2. `transcribe_yt.py` -

#### a. Class: `TranscribeYtVideo`

##### i. Function: `check_yt_cc`

**Description:** A function to check whether the Youtube video has closed captions in English or not.

**Inputs:** None

**Output:** Return textual transcript from closed captions, if present else None

##### ii. Function: `transcribe_yt_video`

**Description:** Check the return of `check_yt_cc` and call further functions according to result

**Inputs:** None

**Output:** returns the summary generated for that video in text format

iii. Function: **transcribe\_yt\_video\_w\_cc**

**Description:** Generate summary for Youtube videos with Closed Captions. Take the closed captions from Youtube video and generate a summary for it.

**Inputs:** None

**Output:** Update the class variable summary with the generated summary

iv. Function: **transcribe\_yt\_video\_wo\_cc**

**Description:** Generate summary for Youtube videos without Closed Captions. Download the video in a temporary location and call transcribe\_video from class TranscribeVideo.

**Inputs:** None

**Output:** Update the class variable summary with the generated summary

3. **summarize.py** -

a. **Class: Summary**

i. Function: **summarize\_text**

**Description:** Generate summary from the provided text.

**Inputs:** Use the class variable transcribed\_text which contains the text from the video

**Output:** Returns the summary generated from the provided text.

4. **split\_audio.py** -

a. **Class: splitwavaudio**

i. Function: **get\_duration**

**Description:** Find the time duration of audio file

**Inputs:** None

**Output:** Returns time of audio in seconds

ii. Function: **single\_split**

**Description:** Create files for every split

**Inputs:**

- from\_min: start time of audio file
- to\_min: end time of audio file
- split\_filename: file name for the temporary audio file

**Output:** Save the temporary audio files created to a temporary location

iii. Function: **multiple\_split**

**Description:** Create splits of large audio files

**Inputs:**

- min\_per\_split: split duration in minutes

**Output:** Returns number of temporary audio files created

5. **cleanup.py** -

a. **Class: Cleanup**

i. Function: **delete\_temp\_files**

**Description:** Delete all the temporary files created during the execution of the entire program.

**Inputs:** None

**Output:** None

6. **scrivener\_user\_interface.py** -

**Description:** Build user interface for the program using Streamlit.

## Test Cases:

### Test Case 1:

Input: Youtube video with cc in English Language. (link = "[Emotional Intelligence: Using the Laws of Attraction | D. Ivan Young | TEDxLSC Tomball - YouTube](#)")

Output:

dr. tom thomas says emotional intelligence isn't just being aware and having clarity about your feelings in the feelings of other people when you're in the midst of a set of circumstances or a situation with them . the truth about emotional intelligence goes far beyond having just clarity about people its clarity about situations circumstances . chimpanzee wearing red tie and a blue and white pinstripe suit . the more emotion you pour into anything the more of that thing you receive . many of us typically spout one thing and do another . we've taken the laws of attraction and mis applied them we don't take serious the things that we feel we deserve enough in order for them to manifest into our lives . if you keep thinking "i'm my Luckies never everybody gets it with me I'll never have that" you end up with more pink slips more bills . everything that is seen comes from that which was not seen that chair you're sitting in the clothes you have on right now with somebody's . everything that we do started out in the unseen and someone called those things that are not as though they are and brought them into manifestation . conscious thinking is not your solution conscious thought is typically reacting to some set of circumstances it's typically paying attention to that which is before you . subconscious mind is the part of you that drives the outcomes in your life as you learn to master being sensitive to circumstances . we get frustrated and angered by that which we do not understand . you have to be willing to take the time in every moment every situation in your life and examine it for its meaning . don't wait to talk truly listen to them you're perceiving it at the expendium . the greatest effort is mastering how do you use your mind as we evolved . we came from somewhere we received of a greater source and conceived . before we get here we made a decision while we were coming . meditate to allow your circumstances to speak to you in such a way that it resonates at the core . if it doesn't make sense to anyon it's time to re-evaluate what you're doing . when you seek the greater truth you get into direct alignment for that very reason that purpose that causes your emotional intelligence and the laws of attraction to become one . if you ever watch a football game you don't see the guy sit there and wait on the ball to come . when a receiver snaps a ball you have to run on faith knowing that there will be the result that you desire and as you feel as you believe you will receive . after all that which you desire is equally seeking you have a great day .

### Test Case 2:

Input: video without cc in English Language (link = [5 Words I'd like to see Retired from Game Discussion - YouTube](#))

Output:

free-to-play games were extremely commonplace about 10 years ago or so . there are quite a few fairly early Asian online games that included pay-to-win as a business model and free to play over there . ost egregious example of pay-to-win as far as i'm concerned . if i wanted to give an example of something that's pretty pay to win then i probably give me example of star war galaxy of heroes . dge most of those accusations being a label it's been applied fake currency thing Galaxy of Heroes there is no currency which allows you to craft a specific hero you have to acquire so cold shots and there are some Heroes who shots don't drop anywhere in the current concept . pay-to-win from classic definition because games have changed business models of change in the definition needs to adapt to that . the pay-to-win aspect of a game can become evident only later down the line . developers have become very very good at hiding it . n is not useful information to the consumer because we all have different definitions . the order 1886 on the frequent use of the term cinematic in various interviews . a Gamespot interview with the developers ready at dawn in February of 2015 reveals occasions in the article however one of these is in reference to the noun it wasn't quotes the Order 1886 . ame rate has no place in video game discussion outside of ps4 . cinematic doesn't apply to video games they apply to mediums . mindless vulnerability has huge periods of time that simply don't have pacing . jeffrey toobin: some places on the internet where you can find a well reasoned discussion about the Cinematic qualities of the Norman interactive components of a video game is any good good . he says it's a very arrogant criticism to make it effectively saying i as an external third-party have become the Arbiter of your feelings . toobin says it is a fruitless Purpose to try and convince people that already really like a product like it . if you wish to criticize the product then include as many facts as possible and then justify a subjective opinion as to why those things aren't good . you can encourage discussion by engaging in constructive criticism that you think you will enjoy . robe like is not a genre but refers to specific characteristics and mechanics of a past genre . as a term it is no longer useful in discussion if you wish to describe a game and situating notes the specific mechanics . g used to have a phrase which is better is beta on the reason I use that little phrase was to say look it's in beta you have to take that into account accepted there might be problems does it mean to call criticizing but you've got to temper that criticism with a reasonable expectation . john avlon: video games are no longer a bay that you released at the very moment that you started charging money for it . he says it is nothing more than a defense used by developers all by apologists for that particular product . the games are almost never truly finished even games that are released on shelves are never really finished .

### Test Case 3:

Input: Youtube video in different Language( link = [How to use an induction stove - YouTube](#))

Output:

```
did a Indochine confirm a rumored indochine rumour? did he confirm that  
he was a man of integrity?
```

### Test Case 4:

Input: Video in .mp4 format (I Was Almost A School Shooter \_ Aaron Stark \_ TEDxBoulder\_144p.mp4)

Output:

```
john sutter: i was almost a school shooter when i got a gun in 1996 . he  
says he figured there was all this extreme emotion going on in his life  
he had to find some way to find control over something .
```

### **Future Scope:**

1. Supports multiple extensions of video files.
2. Summarization of multiple languages.
3. Summarization of Podcasts (audiofiles).
4. Auto summary as a video.
5. Summarization of videos for particular time frames.
6. Compare Summarization models.
7. Improve UI.
8. Summarization as an audio.
9. Summarization of audios for particular time frames.
10. Adding Chrome extension.
11. Sentiment Analysis of the summary.