

Capstone Project

Video and Text Summarizer Updates



Joel

Problem Statement

Students struggle with a lack of time when it comes to reading long articles or watching videos. I wish to help reduce the time it takes to **summarise** said articles/videos.

Input

YouTube URL
Article URL
Readings PDF

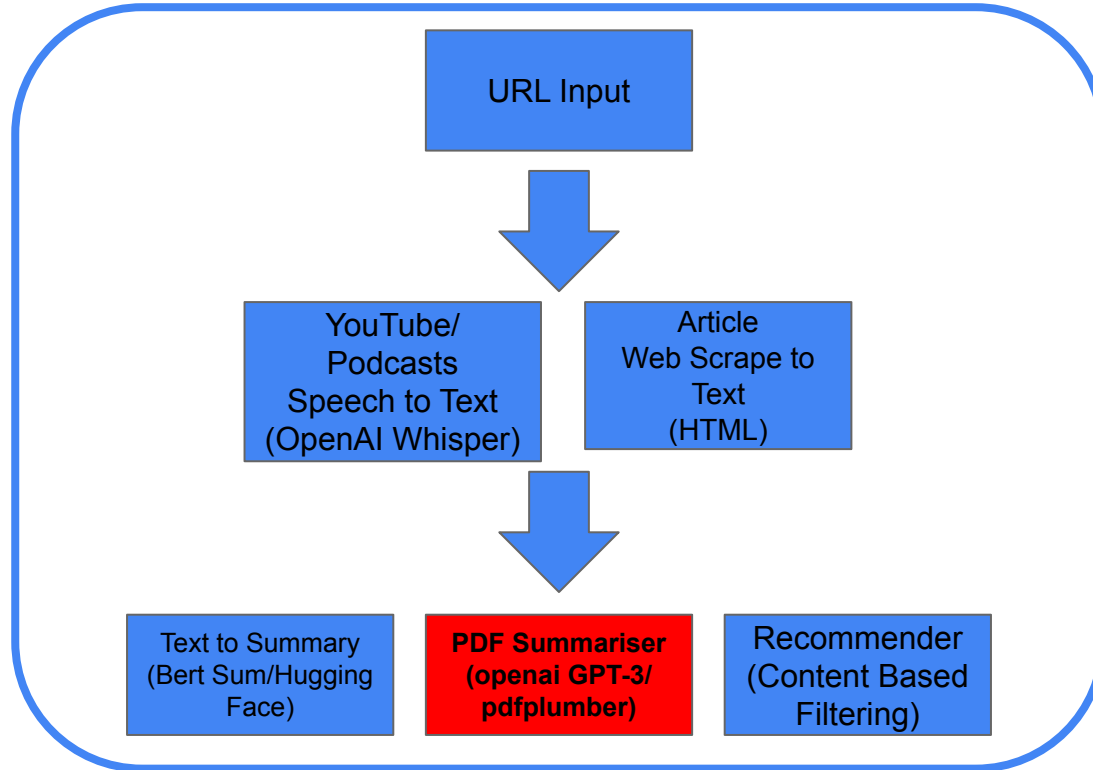


Output

Summary in Text
Format

Project Architecture (New Stuff in Red)

GUI (tkinter)



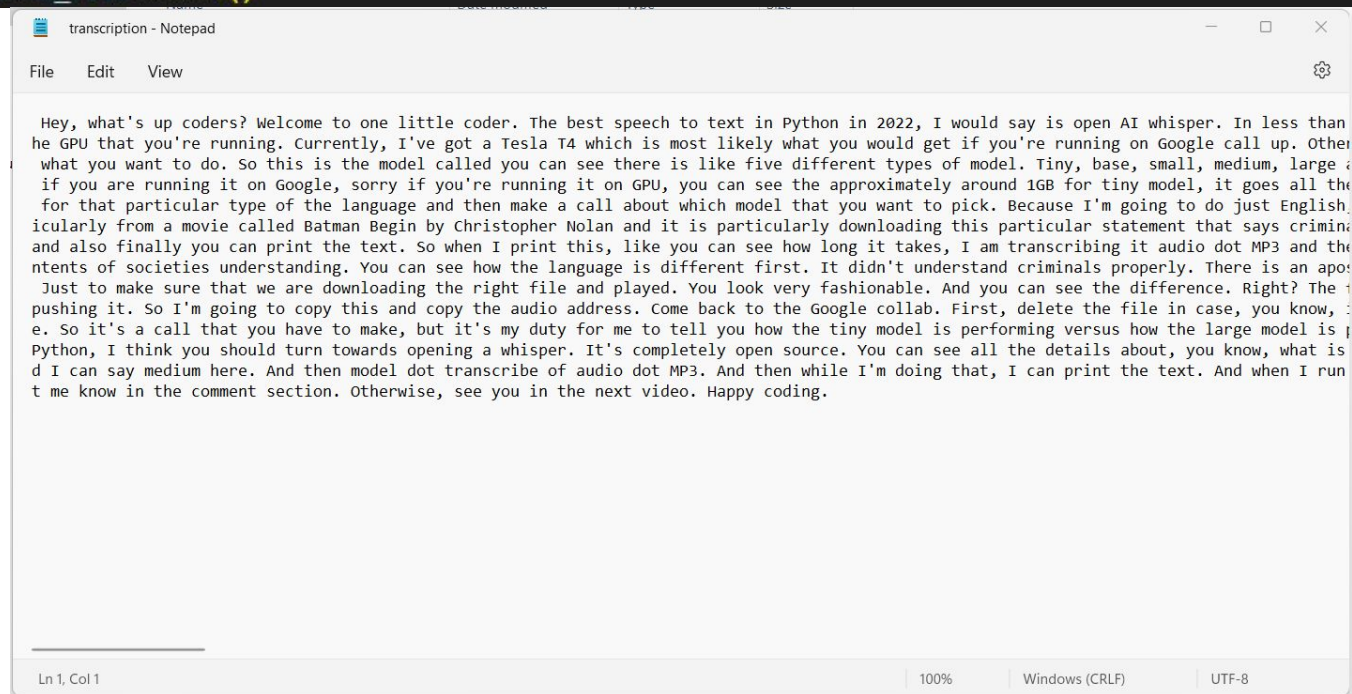
Desktop
Icon
(Pyinstaller)

YouTube Downloader (pytube)

```
1  from pytube import YouTube
2
3  def Download():
4      print("Put your YouTube link here:")
5      link = input()
6      youtubeObject=YouTube(link)
7      youtubeObject=youtubeObject.streams.get_highest_resolution()
8
9      try:
10         youtubeObject.download(filename = "downloaded")
11     except:
12         print("There has been an error in downloading this video")
13
14     print("Download Successful!")
```

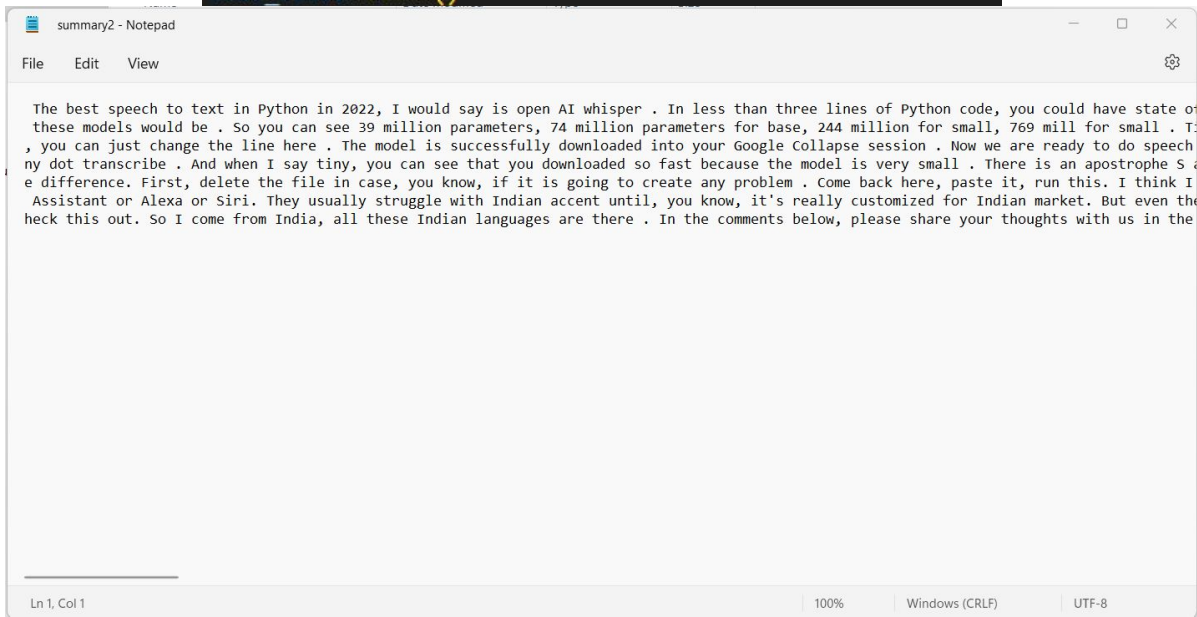
Speech to Text (whisper)

```
model = whisper.load_model('base')
model_text = model.transcribe('3. Lazy Student GUI\downloaded_video', fp16=False)['text']
text_file = open(user_path + r'\transcribed_text.txt', "w")
text_file.write(model_text)
text_file.close()
```

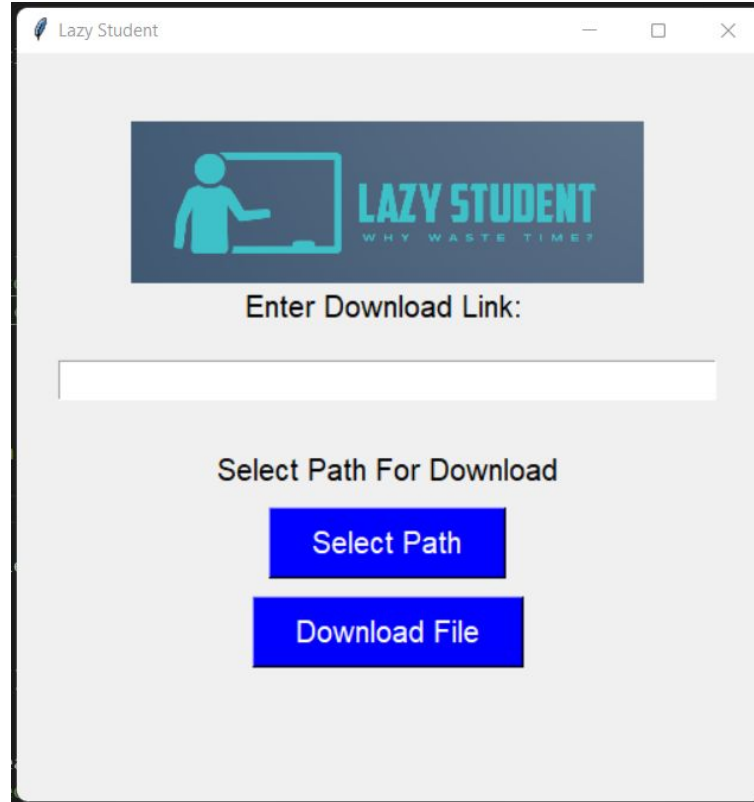


Text to Summary (huggingface, transformers)

```
summarizer = pipeline('summarization')
with open(user_path + r'\transcribed_text.txt') as file:
    lines = file.readlines()
summary_text=summarizer(lines[0])
text_file2 = open(user_path + r"\summary.txt", "wt")
text_file2.write(summary_text[0]['summary_text'])
text_file2.close()
```



NEW: GUI (tkinter)




← URL Input

← File Path to Save
Summary

← Click to Begin
Download

NEW: Desktop Icon (Pyinstaller)

| Name | Date modified | Type | Size |
|---|-------------------|-------------|------------|
|  LazyStudent | 7/2/2023 11:09 pm | Application | 647,685 KB |

Large File

NEW: PDF Summariser (openai)

```
def showPaperSummary(paperContent):  
    tldr_tag = "\n tldr:"  
    openai.organization = 'API KEY org'  
    openai.api_key = "sk-WTwnS3Vv8Yz19o5ub8fhT3B1bkFJZHxEKKjGn05KvzibTYix" #"your openAI key"  
    engine_list = openai.Engine.list()  
  
    for page in paperContent:  
        text = page.extract_text() + tldr_tag  
        response = openai.Completion.create(engine="davinci",prompt=text,temperature=0.3,  
            max_tokens=140,  
            top_p=1,  
            frequency_penalty=0,  
            presence_penalty=0,  
            stop=["\n"]  
        )  
        print(response["choices"][0]["text"])  
  
paperContent = pdfplumber.open(r"C:\Users\redoc\OneDrive\Desktop\Sociology-SQP.pdf").pages  
showPaperSummary(paperContent)
```

Still a work in progress

Challenges

- Huggingface Summariser too slow. Working on using OpenAI GPT Library.
- Pyinstaller's desktop icon is inconsistent, sometimes it does not run the program.

Thank you :)