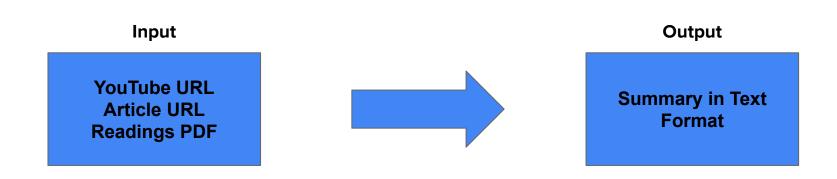
Capstone Project

Video and Text Summarizer Updates

Joel

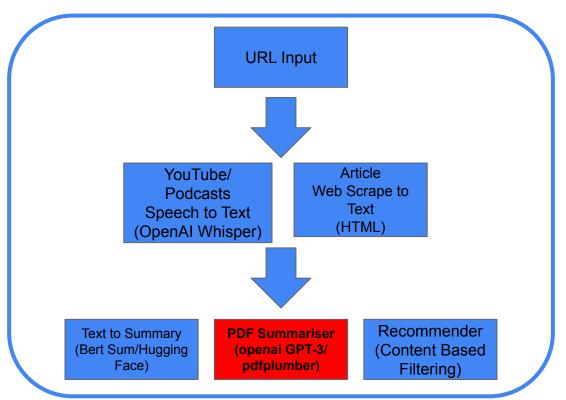
Problem Statement

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.



Project Architecture (New Stuff in Red)

GUI (tkinter)



Desktop Icon (Pyinstaller)

YouTube Downloader (pytube)

```
from pytube import YouTube
def Download():
    print("Put your YouTube link here:")
    link = input()
    youtubeObject=YouTube(link)
    youtubeObject=youtubeObject.streams.get highest resolution()
    try:
        youtubeObject.download(filename = "downloaded")
    except:
        print("There has been an error in downloading this video")
    print("Download Successful!")
```

Speech to Text (whisper)

Edit

View

```
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()
```

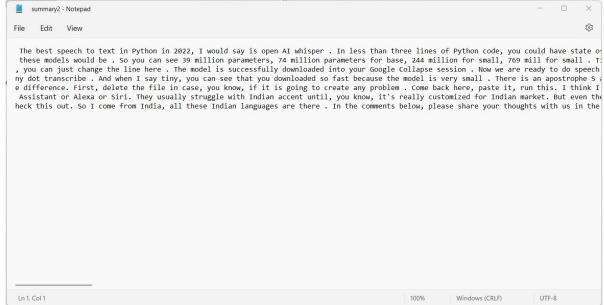
Hey, what's up coders? Welcome to one little coder. The best speech to text in Python in 2022, I would say is open AI whisper. In less than he GPU that you're running. Currently, I've got a Tesla T4 which is most likely what you would get if you're running on Google call up. Other what you want to do. So this is the model called you can see there is like five different types of model. Tiny, base, small, medium, large a if you are running it on Google, sorry if you're running it on GPU, you can see the approximately around 1GB for tiny model, it goes all the for that particular type of the language and then make a call about which model that you want to pick. Because I'm going to do just English icularly from a movie called Batman Begin by Christopher Nolan and it is particularly downloading this particular statement that says crimina and also finally you can print the text. So when I print this, like you can see how long it takes, I am transcribing it audio dot MP3 and the network of societies understanding. You can see how the language is different first. It didn't understand criminals properly. There is an apport to make sure that we are downloading the right file and played. You look very fashionable. And you can see the difference. Right? The pushing it. So I'm going to copy this and copy the audio address. Come back to the Google collab. First, delete the file in case, you know, e. So it's a call that you have to make, but it's my duty for me to tell you how the tiny model is performing versus how the large model is python, I think you should turn towards opening a whisper. It's completely open source. You can see all the details about, you know, what is d I can say medium here. And then model dot transcribe of audio dot MP3. And then while I'm doing that, I can print the text. And when I run t me know in the comment section. Otherwise, see you in the next video. Happy coding.

(3)

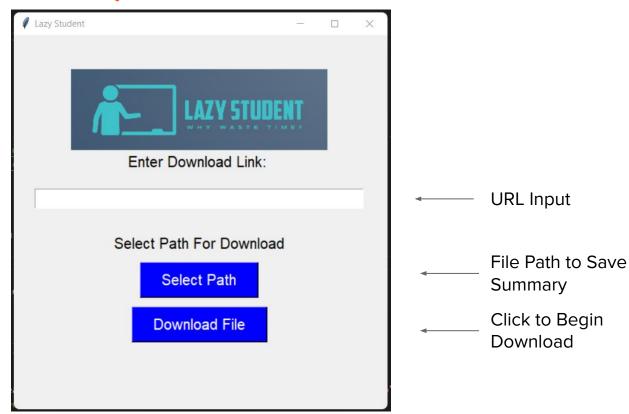
Ln 1, Col 1 100% Windows (CRLF) UTF-8

Text to Summary (huggingface, transformers)

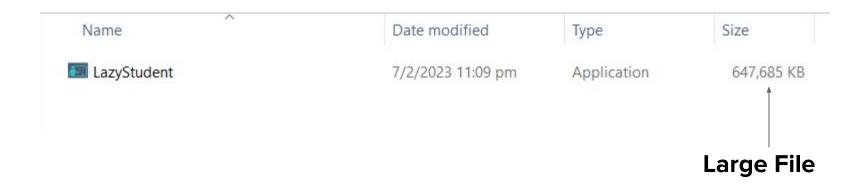
```
summarizer = pipeline('summarization')
with open(user_path + r'\transcribed_text.txt') as file:
    lines = file.readlines()
summary_text=summarizer(lines[0])
text_file2 = open(user_path + r"\summary.txt", "wt")
text_file2.write(summary_text[0]['summary_text'])
text_file2.close()
```



NEW: GUI (tkinter)



NEW: Desktop Icon (Pyinstaller)



NEW: PDF Summariser (openai)

```
def showPaperSummary(paperContent):
    tldr tag = "\n tl;dr:"
    openai.organization = 'API KEY org'
    openai.api key = "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" #"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-SOP.pdf").pages
showPaperSummary(paperContent)
```

Challenges

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

Thank you:)