

# **Capstone Project**

## **Video and Text Summarizer**

### **Final Presentation**



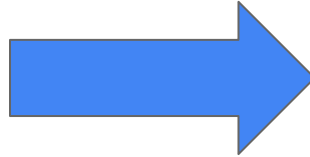
Joel Quek

# 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

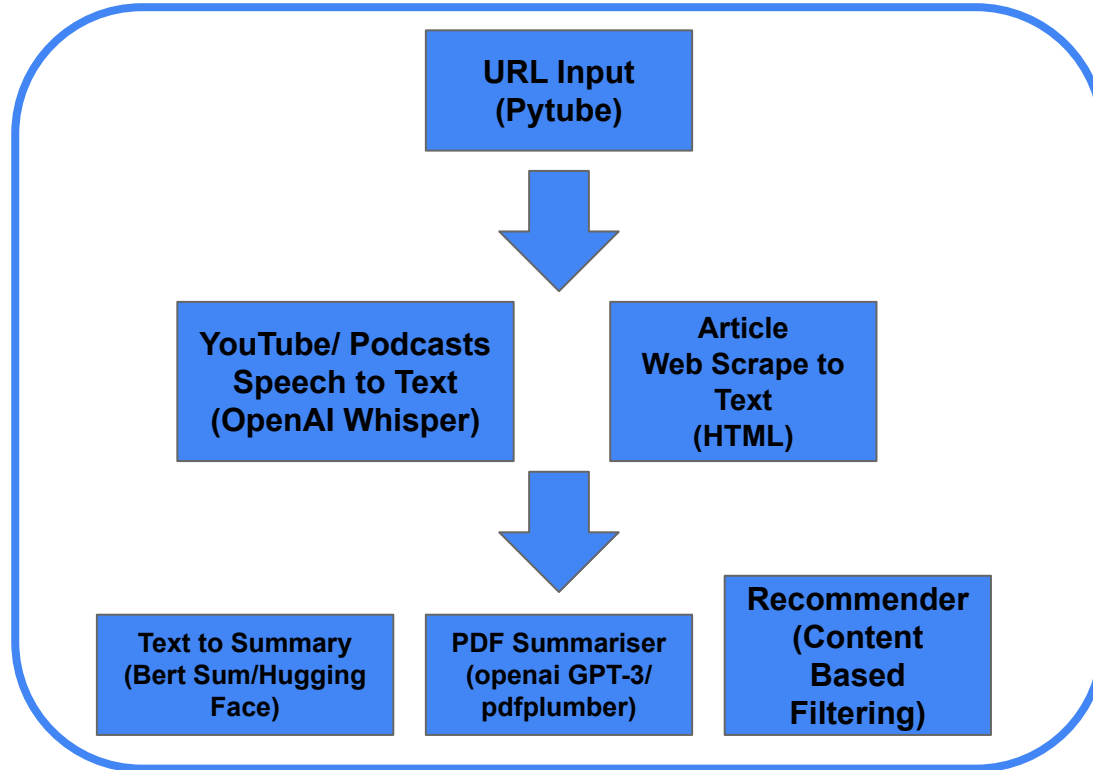


# LAZY STUDENT

W H Y   W A S T E   T I M E ?

# Project Architecture [Broad Strokes]

GUI (tkinter)



Desktop  
Icon  
(Pyinstaller)

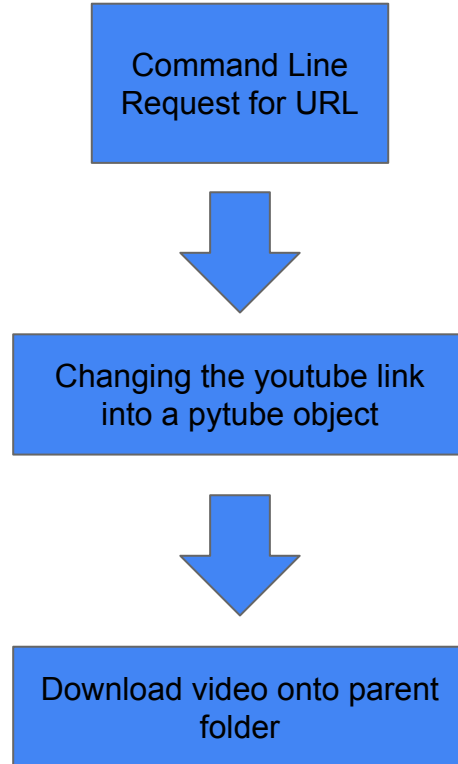
# **“Objects” and “Encapsulation”**

Every node/stage in the project architecture can be seen as an “object”.

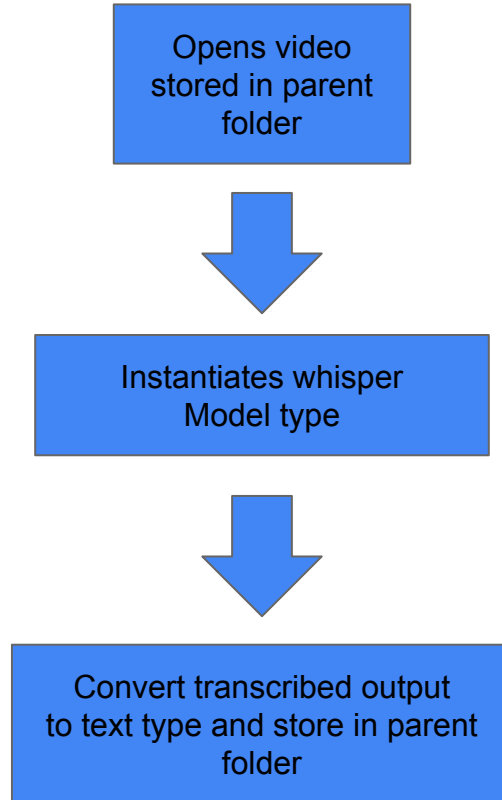
“Encapsulation” in the form of the GUI.

But not exactly OOP.

# YouTube Downloader [pytube]



# Speech to Text [whisper by openai]



# Whisper Model Types and Speed

Source: <https://github.com/openai/whisper>

Model Type is now included in the GUI Selections

Size	Parameters	English-only model	Multilingual model	Required VRAM	Relative speed
tiny	39 M	<code>tiny.en</code>	<code>tiny</code>	~1 GB	~32x
base	74 M	<code>base.en</code>	<code>base</code>	~1 GB	~16x
small	244 M	<code>small.en</code>	<code>small</code>	~2 GB	~6x
medium	769 M	<code>medium.en</code>	<code>medium</code>	~5 GB	~2x
large	1550 M	N/A	<code>large</code>	~10 GB	1x



# Whisper Streamlining

Video **Example:** <https://www.youtube.com/watch?v=NiKtZglmdlY>

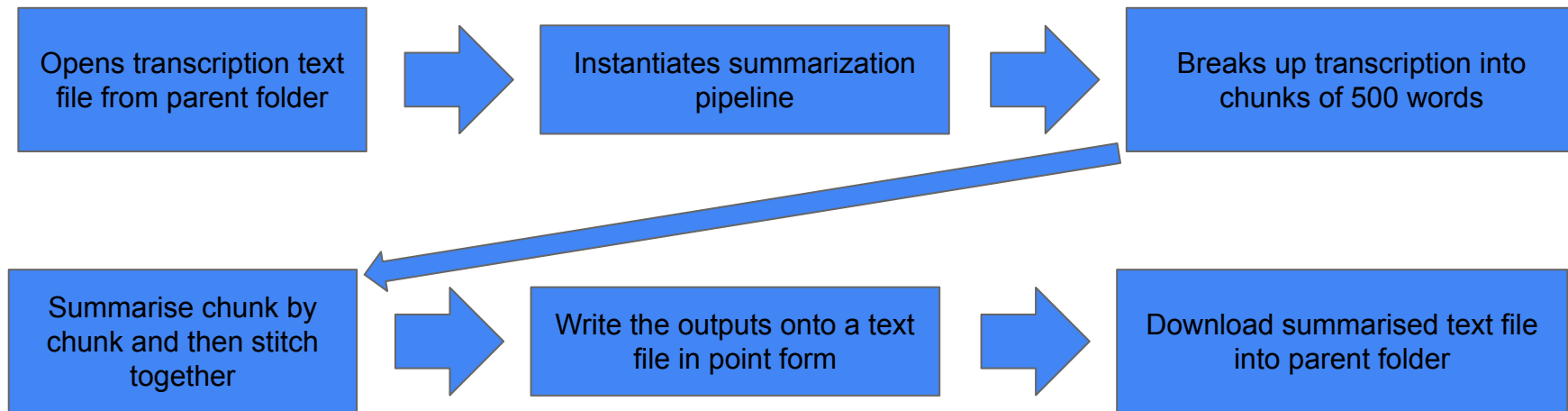
Video Length: 4 min 22s

Model	Tiny	Base
Video Download	5 seconds	11 seconds
Transcription	36 seconds	55 seconds
Summary	1 min 33s	1 min 47s
Transcription Length	750 words	753 words
Summary Length	452 words	442 words

# The Summarizer [huggingface]






# Text to Summary [huggingface, transformers]



```
for i in range(len(list)):
    text_file2.write("-"+list[i])
    text_file2.write("\n")
    text_file2.write("\n")
text_file2.close()
shutil.move(text_file2, user_path)
```

# Huggingface Pipelines

Models	Examples
Text-classification	<pre>sequence_to_classify = "one day I will see the world" candidate_labels = ['travel', 'cooking', 'dancing'] classifier(sequence_to_classify, candidate_labels)</pre> <p>Could be used for <b>recommender systems</b></p>
Roberta-large-mnli	
Sentiment analysis	 positive,  negative, or  neutral

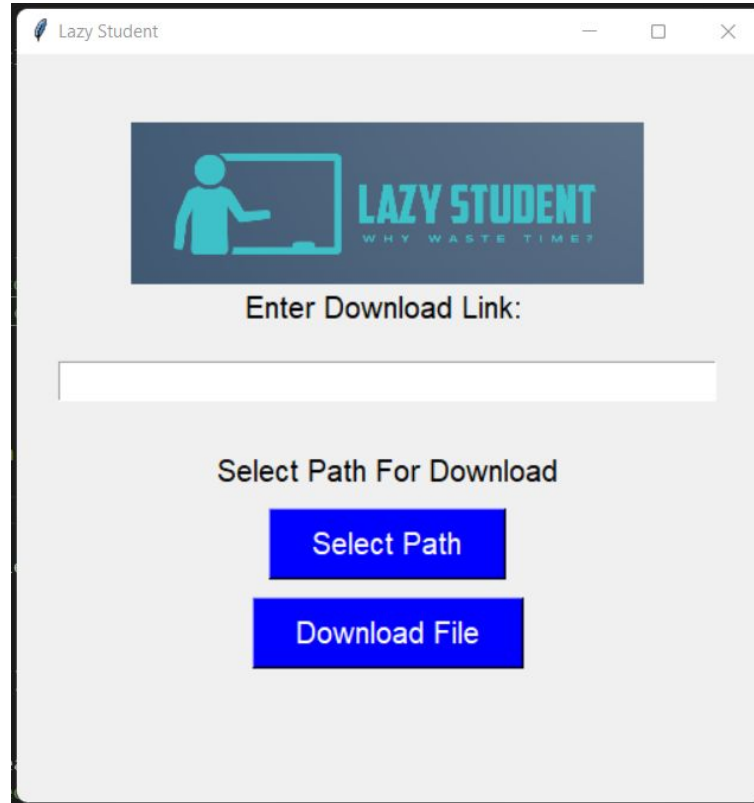
# Huggingface Streamlining

Problem	Solution
Huggingface cannot summarise large chunks (>500 words)	Break down the transcription file into chunks of 500 words
Too many pipelines to choose from	Just use summarisation because that is the output I want
Huggingface quite slow	<b>GPT3 API</b>

# The GUI [tkinter]



# GUI Version 1 [tkinter]



← URL Input

← File Path to Save  
Summary

← Click to Begin  
Download

# GUI Version 2 [tkinter]

Model Type  
[New]



The screenshot shows a Tkinter window titled "Summarising Complete!". At the top is a logo for "LAZY STUDENT" with the tagline "WHY WASTE TIME?". Below the logo, the text "Enter Download Link:" is followed by a text input field containing the URL "https://www.youtube.com/watch?v=NiKtZglmdlY". Underneath the URL field, the file path "C:/Users/redoc/OneDrive/Desktop" is displayed. A blue button labeled "Select Path" is positioned below the path. Further down, the text "Enter Model Type [tiny, base or small]:" is followed by a dropdown menu currently set to "tiny". At the bottom of the window is a large blue button labeled "Download File".

← URL Input

← File Path to Save  
Summary (Parent  
Folder)


← Click to Begin  
Download



# **Expansion Pack**

## **[Sometimes can. Sometimes Cannot]**

# Desktop Icon [Pyinstaller]

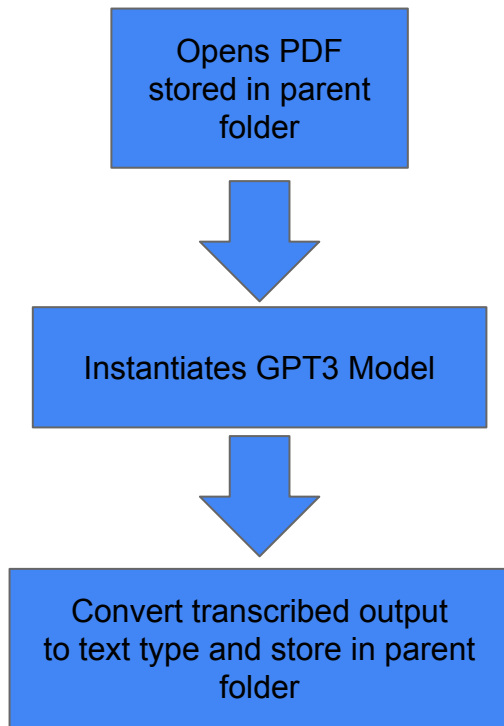
Name	Date modified	Type	Size
 LazyStudent	7/2/2023 11:09 pm	Application	647,685 KB

**Large File**

Still a work in progress



# PDF Summariser (openai)



Still a work in progress



# Thank you :)

