

EXTRACT TEXT FROM VIDEO

As a project work for Course

PYTHON PROGRAMMING (INT 213)

Submitted By:

Name : Varada Sai Vamsi

Section : K20bg

Reg No : 12006609

Roll No : A23

Program- B. Tech CSE

Semester- Third

Submitted To:

Name : Dr. Sagar Pande

Professor



Lovely Professional University

Jalandhar, Punjab , India

DATE OF SUBMISSION: 20th NOVEMBER 2021

CONTENT:

Serial no	Name	Page no:
1.	ABSTRACT	3
2.	INTRODUCTION	3
3.	OBJECTIVE	4
4.	DATA FLOW DIAGRAM	4-5
5.	RELATED WORK	6-7
6.	METHODOLOGY	7-10

7.	RESULT ANALYSIS	10
8.	ADVANTAGES	10
9.	DISADVANTAGES	10
10.	FUTURE SCOPE	10
11.	CONCLUSION	10
12.	REFERENCES	11

1.ABSTRACT:

Now a days, we may prefer to watch a film in OTT, but we may not understand foreign languages and for that film we may feel bored to see if the film doesn't present subtitles. Suppose one would love to watch a Hollywood movie or any other film rather than our mother tongue, which we don't understand that we definitely prefer subtitles to know what they speak so, as result here I have made a simple speech extractor from video which will recognize "English" language and converts it into text.

2.INTRODUCTION:

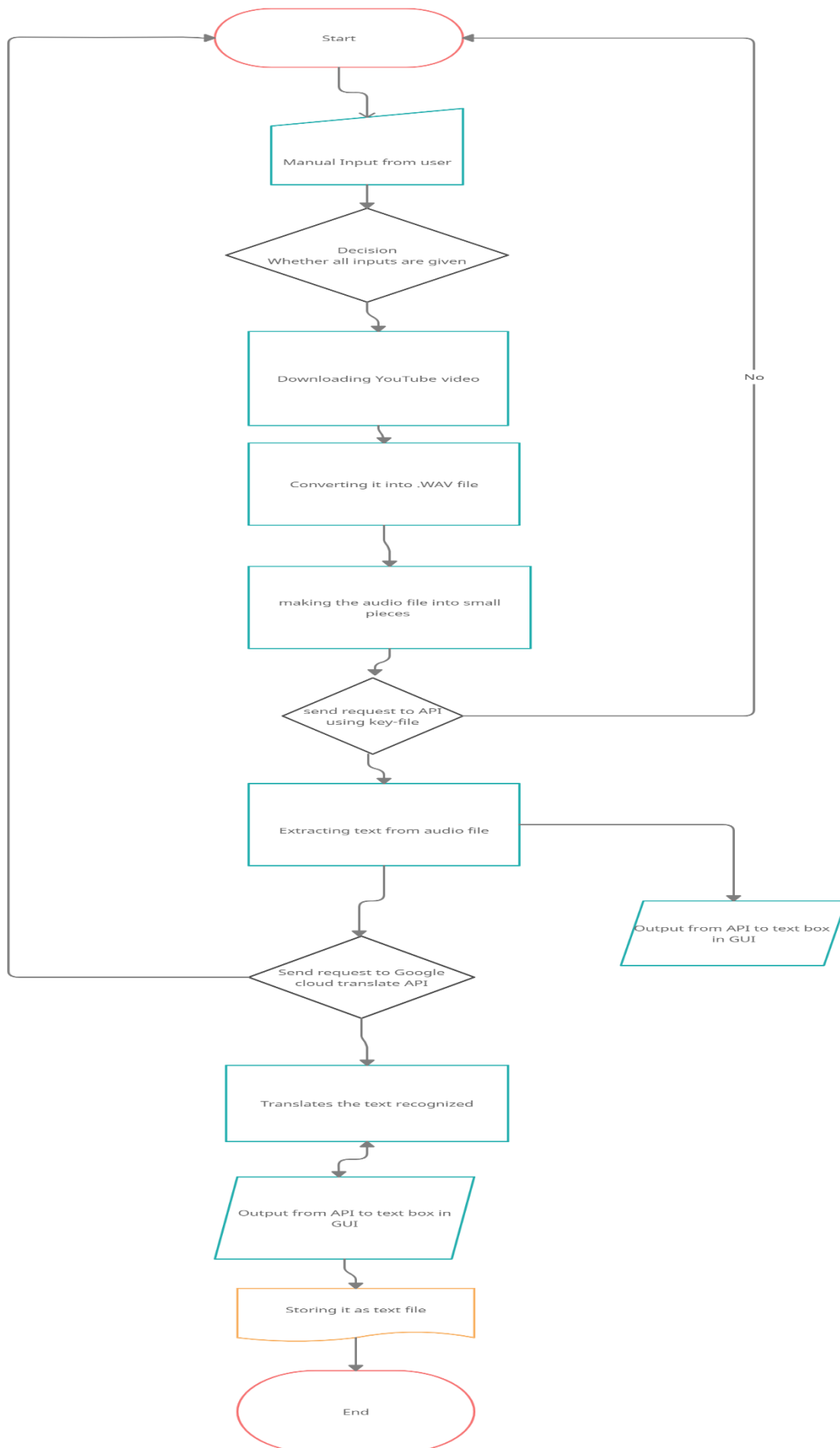
Speech recognition is an interesting task that allows you to improve the quality of your life. In this never ending Covid period, I need to watch many videos of lessons, and it's so easy to lose concentration. At the same time, the possibility to have all registrations available on my university's website made me become a perfectionist, so I would like to take every word in my notes. But it's costly because it needs a lot of work and steals time.

3.OBJECTIVE:

The AIM of the project is to Download a YouTube video and extract its speech to text and translate it into desired language using a Graphical User Interface.

4.DATAFLOW DIAGRAM

EXTRACT TEXT FROM VIDEOS



5.RELATED WORK:

S.N	NAME	USE	URL
0			
1.	TOWARDS AI	Speech recognition	https://towardsai.net/p/nlp/extract-the-text-from-long-videos-with-python
2.	GeekForGeeks	YouTube video download	https://www.geeksforgeeks.org/how-to-extract-youtube-data-in-python/
3.	The Python Code	Speech recognition	https://www.thepythoncode.com/article/using-speech-recognition-to-convert-speech-to-text-python
4.	GeekForGeeks	GUI	https://www.geeksforgeeks.org/python-gui-tkinter/
5.	javatpoint	GUI	https://www.javatpoint.com/python-tkinter
6.	tutorialspoint	GUI	https://www.tutorialspoint.com/gui_programming_with_python_tkinter_and_java_swing/index.asp
8.	Stack Overflow	YouTube video download	https://stackoverflow.com/questions/40713268/download-youtube-video-using-python-to-a-certain-directory/54163735
9.	Google Cloud	Speech to text API	https://cloud.google.com/speech-to-text/docs/quickstart-client-libraries
10.	Google Cloud	Translate API	https://cloud.google.com/translate/docs

These are some websites which I have used to make my project differently .

I have created a Graphical User Interface in my program so a normal person can interact with my code easily.

Some Libraries which are used in project:**Google Cloud Speech to text API:**

The Cloud Speech API enables developers to convert audio to text by applying powerful neural network models. The API recognizes over 80 languages and variants, to support your global user base.

Google Cloud Translate API:

With Google Cloud Translation, you can dynamically translate text between thousands of language pairs. The Google Cloud Translation API lets websites and programs integrate with Google Cloud Translation programmatically.

MoviePy:

MoviePy is a Python library for video editing: cutting, concatenations, title insertions, video compositing (a.k.a. non-linear editing), video processing, and creation of custom effects.

MoviePy can read and write all the most common audio and video formats, including GIF, and runs on Windows/Mac/Linux, with Python 2.7+ and 3 (or only Python 3.4+ from v.1.0).

Pydub:

Audio files are a widespread means of transferring information. So, let's see how to work with audio files using Python. Python provides a module called **pydub** to work with audio files. **pydub** is a Python library to work with **only .wav** files. By using this library, we can play, split, merge, edit our , .wav audio files.

6.METHODOLOGY:

Step 1:

Install all libraries required

- pip install --upgrade google-cloud-speech
- pip install google-cloud-translate==2.0.1
- pip install moviepy
- pip install pydub
- pip install pytube

Step 2:

Importing required libraries

```
import tkinter as tk
from tkinter import *
import tkinter.font
import pytube
import moviepy.editor as mp
from pydub import AudioSegment
from pydub.silence import split_on_silence
from google.cloud import speech
import os
import io
```

EXTRACT TEXT FROM VIDEOS

Step 3:

Downloading a YouTube video using its URL.

```
link=var.get()
try:
    yt = pytube.YouTube(link)
    video=yt.streams.filter(progressive=True, file_extension='mp4').order_by('resolution')[-1]
    video.download(var1.get(),var2.get())
except OSError as error:
    lberror=Label(root,text=error)
    lberror.configure(font=Desired_font)
    lberror.place(x=30,y=500)
except:
    lb5=tk.Label(root,text='error in connection',bg=rgb_hack((204, 255, 204)))
    lb5.configure(font=Desired_font)
    lb5.place(x=30,y=500)
```

Step 4:

Converting it into .WAV file

```
def wavConversion():
    try:
        dire=var1.get()+"\\"+var2.get()
        print(dire)
        clip = mp.VideoFileClip(dire)
        direC=var1.get()+"\\"+var2.get().split('.')[0]+'wav'
        clip.audio.write_audiofile(direC)

        lb2=tk.Label(root,text='video downloaded and extracted successfully',bg=rgb_hack((204, 255, 204)))
        lb2.configure(font=Desired_font)
        lb2.place(x=30,y=500)
        text=get_large_audio_transcription(direC)
        text1.insert(END,text)
        text1.insert(END,"\n")
        text1.insert(END,"\n")
        text1.insert(END,"\n")
        translation=translate_text(var4.get(),text)
        text1.insert(END,"Translated text is: ")
        text1.insert(END,translation)
        counting=len(text.split())
        fileD=var1.get()+"\\"+var2.get().split('.')[0]+'txt'
        file=open(fileD,'a')
        file.write(text)
        file.write("\n\n\n")
        file.close()
        count2=Label(root,text=str(counting))
        count2.configure(font=Desired_font)
        count2.place(x=900,y=575)
```

Step 5:

Sending request to Google cloud from client side by using our key-file and getting our output.

EXTRACT TEXT FROM VIDEOS

```
def get_large_audio_transcription(path):
    sound = AudioSegment.from_wav(path)
    chunks = split_on_silence(sound,
        min_silence_len = 500,
        silence_thresh = sound.dBFS-14,
        keep_silence=500,
    )
    folder_name = "audio-chunks"
    if not os.path.isdir(folder_name):
        os.mkdir(folder_name)
    whole_text = ""
    for i, audio_chunk in enumerate(chunks, start=1):
        chunk_filename = os.path.join(folder_name, f"chunk{i}.wav")
        audio_chunk.export(chunk_filename, format="wav")
        client = speech.SpeechClient()
        with io.open(chunk_filename, "rb") as audio_file:
            content = audio_file.read()
            audio = speech.RecognitionAudio(content=content)
        config = speech.RecognitionConfig(
            encoding=speech.RecognitionConfig.AudioEncoding.LINEAR16,
            audio_channel_count=2,
            language_code="en-US",
        )
        try:
            response = client.recognize(request={"config": config, "audio": audio})

            for result in response.results:
                print("Transcript: {}".format(result.alternatives[0].transcript))
                whole_text=whole_text+result.alternatives[0].transcript
        except:
            errorLabel=Label(root,text='An error occurred during recognition',bg=rgb_hack((204, 255, 204)))
            errorLabel.configure(font=Desired_font)
            errorLabel.place(x=30,y=500)

    print(whole_text)
```

Step 4:

Translating it into target language

```
def translate_text(target, text):
    """Translates text into the target language.

    Target must be an ISO 639-1 language code.
    See https://g.co/cloud/translate/v2/translate-reference#supported_languages
    """

    import six
    from google.cloud import translate_v2 as translate
    translate_client = translate.Client()
    if isinstance(text, six.binary_type):
        text = text.decode("utf-8")
    result = translate_client.translate(text, target_language=target)
    print(u"Text: {}".format(result["input"]))
    print(u"Translation: {}".format(result["translatedText"]))
    print(u"Detected source language: {}".format(result["detectedSourceLanguage"]))
    return result['translatedText']
```

By using Google cloud translate API by sending request using our client key-file.

Step 5:

Making it available as .txt file in the directory where video has installed in the user end.

The Complete Code Link:

<https://github.com/vamsivarada3008/extract-text-from-videos/blob/branch-1/extractTextFromVideo>

7.RESULT ANALYSIS:

As this project is having a GUI so that user cannot have any struggle in downloading ,extracting and translating it.

Here in this project, I have used Google cloud speech to text and Google cloud Translate API so there are less chances in having mistakes as it uses powerful neural algorithms to extract text in speech.

8.ADVANTAGES:

There are so many advantages in this project

- User can easily get translation of the video
- User can easily get speech in the video so that he may understand well.
- It is very helpful in understanding foreign languages
- It is useful in storing YouTube videos in our device and use it for future for learning purpose.

9.DISADVANTAGES:

Here using APIs may make our job easier but

- The user should have an account in google cloud and he have to enable APIs for usage purpose
- And based on the usage of API user may have to pay for it.

10.FUTURE SCOPE:

- This is a simple Machine Learning project which deals with speech to text recognition module and translation module
- Learning M L engineering we get jobs in MNCs ,because training our system to do task will makes job easier .And in present days there is big demand for ML and it continues in future .

11.CONCLUSION:

Hence, with the help of pydub, moviepy and Google Cloud Speech API we made this project there are several recognition APIs like CMU Sphinx (offline) , , Wit.ai , Microsoft Bing Voice Recognition, Houndify API, IBM Speech To Text, Snowboy Hotword Detection (offline) etc these are very helpful APIs which makes our job so easier and unlike all google cloud speech API is very powerful API and using this we not only recognize speech but also we can make subtitle folder i.e. .srt file and insert it in required video and also we can convert the recognized speech into other language and use it in our project.

12.REFERENCES:

1. <https://towardsai.net/p/nlp/extract-the-text-from-long-videos-with-python>
2. <https://www.geeksforgeeks.org/how-to-extract-youtube-data-in-python/>
3. <https://www.thepythoncode.com/article/using-speech-recognition-to-convert-speech-to-text-python>
4. <https://www.geeksforgeeks.org/python-gui-tkinter/>
5. <https://www.javatpoint.com/python-tkinter>
6. https://www.tutorialspoint.com/gui_programming_with_python_tkinter_and_java_swing/index.asp
7. <https://stackoverflow.com/questions/40713268/download-youtube-video-using-python-to-a-certain-directory/54163735>
8. <https://cloud.google.com/speech-to-text/docs/quickstart-client-libraries>
9. <https://cloud.google.com/translate/docs>