Assignment 2 – Subtask 1

Name: AASTHA A K VERMA

Entry Number: 2022CS11607

Course: COP290

# 1 Video/Audio Processing

Our objectives are attained using Python because of the ease of use and availability of a variety of libraries for different functionalities.

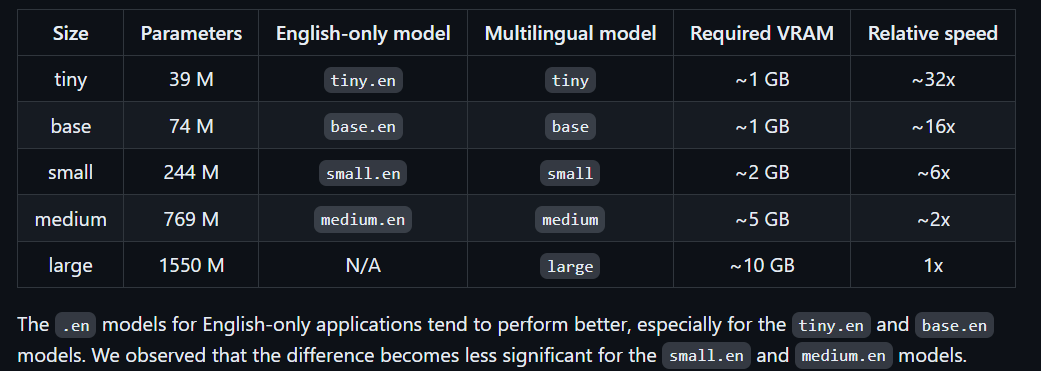
## 1.1 Workflow

The following workflow is followed:

## 1.2 Experiments and Challenges

In the process to find a free Python library suited for speech recognition, speechrecognition, speechbrain and whisper libraries were tested.

* speechrecognition supports several APIs but most of them are paid or with limited free usage. The recognize\_google function calls the old Google Speech API which is no longer maintained, and the result was not good. For some tests, parts of the actual speech were missing in the reported text. The API also failed with the introduction of some high-tempo music beats and reported the speech as “unintelligible”, raising an UnknownValueError.
* speechbrain was difficult to implement because of the requirement for several dependencies like symlink, and some technicalities like administrator privileges. It can probably be explored along with the HuggingFace-hosted models for more complex applications, but it seemed to be overkill for our purposes.
* OpenAI’s whisper is open-source and produces excellent, almost correct speech-to-text. It is also trained to recognize sounds like music and humming. It supports multiple languages, with automatic detection. It supports variations of models, “medium” has been selected here.



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

Hence, whisper generated the best results and was finally used.

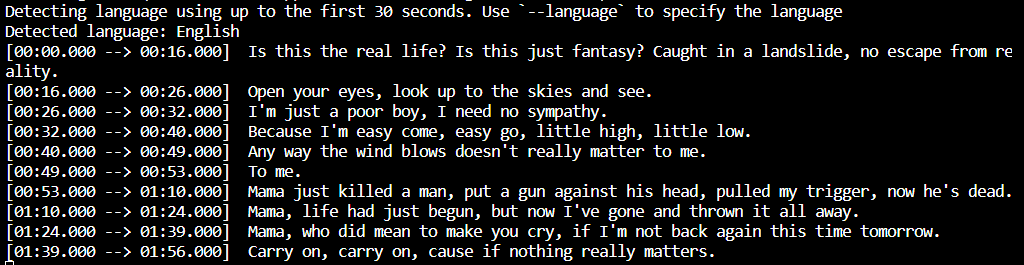
## 1.3 Testing

The sample videos are converted to audio to test both functionalities simultaneously. The samples are selected to test the following criteria:

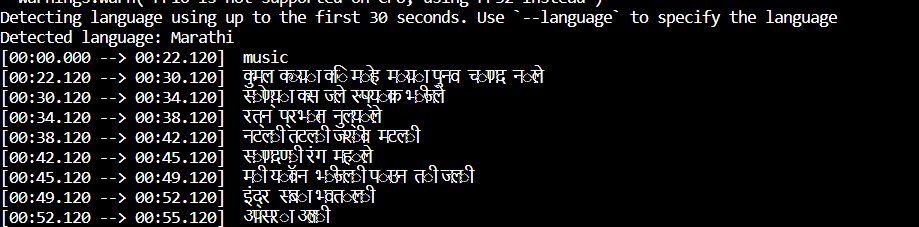
* Media duration should not affect the result.
* Proper recognition for speech that is not conversational, e.g., songs with heavy instrumental influence.
* Media possibly not being in English.
* Efficient filtering of possible background noise/voices.

In this light, the selected samples consist of normal conversational-style videos, narratives, clearly-enunciated songs, some poorly-enunciated songs, some instrumentally heavy songs, and a vernacular clip in Marathi.

Marking verbose=True in the transcribe() function displays the output of the sliding window at the terminal.



The Marathi video’s language is correctly recognised.



# 2 PDF Processing

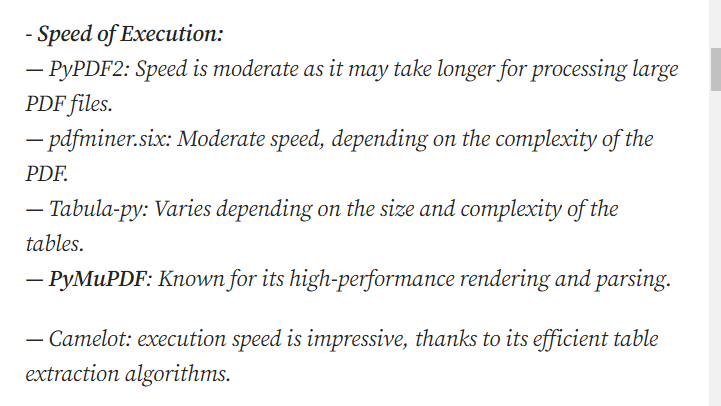
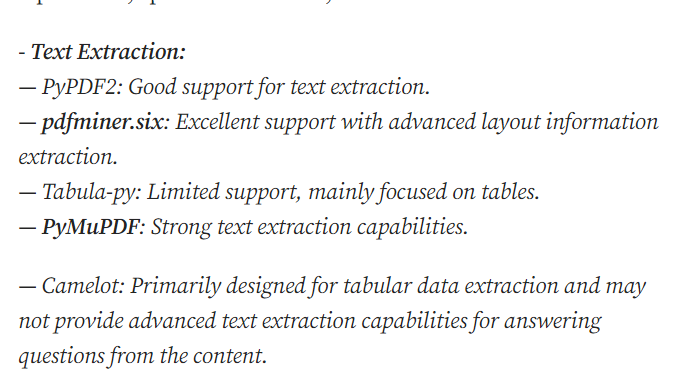
Our objectives are attained using Python because of the ease of use and availability of a variety of libraries for different functionalities.

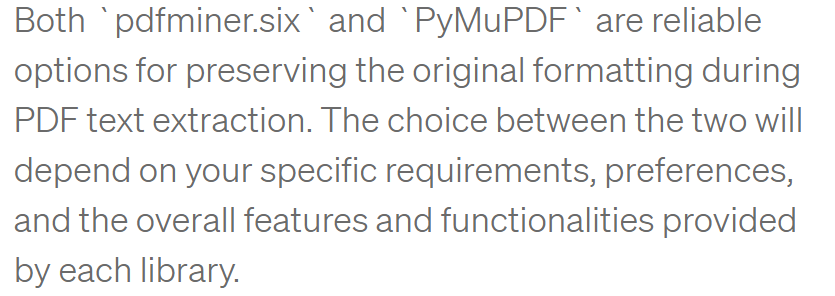
## 2.1 Experiments and Challenges

This part was relatively less challenging because of the availability of a variety of libraries. So multiple libraries were implemented with the differences in their result mostly being the decision to put in newlines. However, benchmarks with large datasets present the following data:



Source: <https://github.com/py-pdf/benchmarks/tree/main/read>





Source: <https://pradeepundefned.medium.com/a-comparison-of-python-libraries-for-pdf-data-extraction-for-text-images-and-tables-c75e5dbcfef8>

Finally, the following libraries were implemented:

* pymupdf
* pypdfium
* pypdf2
* pdfminer.six
* tika (Note that Tika is written in Java so we need a Java runtime installed.)
* tabula (for an exclusively table-containing pdf)

## 2.2 Testing

The pdfs were selected to test the following parameters:

* Text extraction quality
* Format preservation
* Text extraction from images in pdf
* Treatment of tables

The sample pdf set consisted of a normal text pdf, a pdf with only images, a pdf with blank underscored lines, pdfs with fancy formatting of text, and a pdf with a table. Text extraction is mostly similar for most libraries, with poor recognition of text from images. tabula gives excellent table comprehension with data being possibly manipulated using Pandas DataFrames and being written to CSV format.

# 3 Results

<https://drive.google.com/drive/folders/1YK2OjW59dYu2HSRBS0gaOGVsOYJWRrU0?usp=sharing>

Follow the above link to view the results of the text-extraction processes. Note that the audio samples are audio extractions of the videos themselves, found in the gen\_audios folder, other samples are not taken.