Iteration 2

Team Name: Codezilla

Team Members:

1. Scrum Master & Developer: Rohan Mahesh Jagiasi
2. Product Owner & Developer: Aishwarya Teegulla
3. Developer: Sunil Krishna Kumar Komadam
4. Developer: Sanjiti Bhargava
5. Developer: Spoorthy Kanduri
6. Developer: Zaid Pervaiz Bhat
7. Developer: Samiksha Marne

Customer/Client Meeting: 28-April-2021 4:30-5:15pm.

Customer/Client Interactions: Zoom Meetings:

Meeting 1: April 7th, 2021
Time: 4:15-5:00pm
Zoom Video Recording:
https://drive.google.com/file/d/1i4PA_DTs95A2XII9OgbvI2tnWHx2Y7dt/view?usp=sharing

Meeting 2: April 13th, 2021
Time: 4:15-5:00pm
Zoom Video Recording:
https://drive.google.com/file/d/1wvgtwXFUtHjIVjcIESBdSj_Ctp53U5VG/view?usp=sharing

Meeting 3: April 20th, 2021
Time: 4:15-5:00pm
Zoom Video Recording:
https://drive.google.com/file/d/15LYUEf1I2PUWoQhOpw0pjrA5XOPjlS69/view?usp=sharing

Story implemented:

- Feature: Apply the models involved in audio/video processing to classify language/physical groups.
  - So that we can get a desired output which classifies the given data
- Feature: Perform post-processing analytics to the outcome of the ML model to provide relevant results as feedback
  - So that we provide output understood and relevant to the client
  - We want to analyse the results from the ML model and process it as expected in TBOP

Design diagram for this iteration:

The pipeline was designed in this iteration. Individual modules of the code were integrated to form a cohesive system.

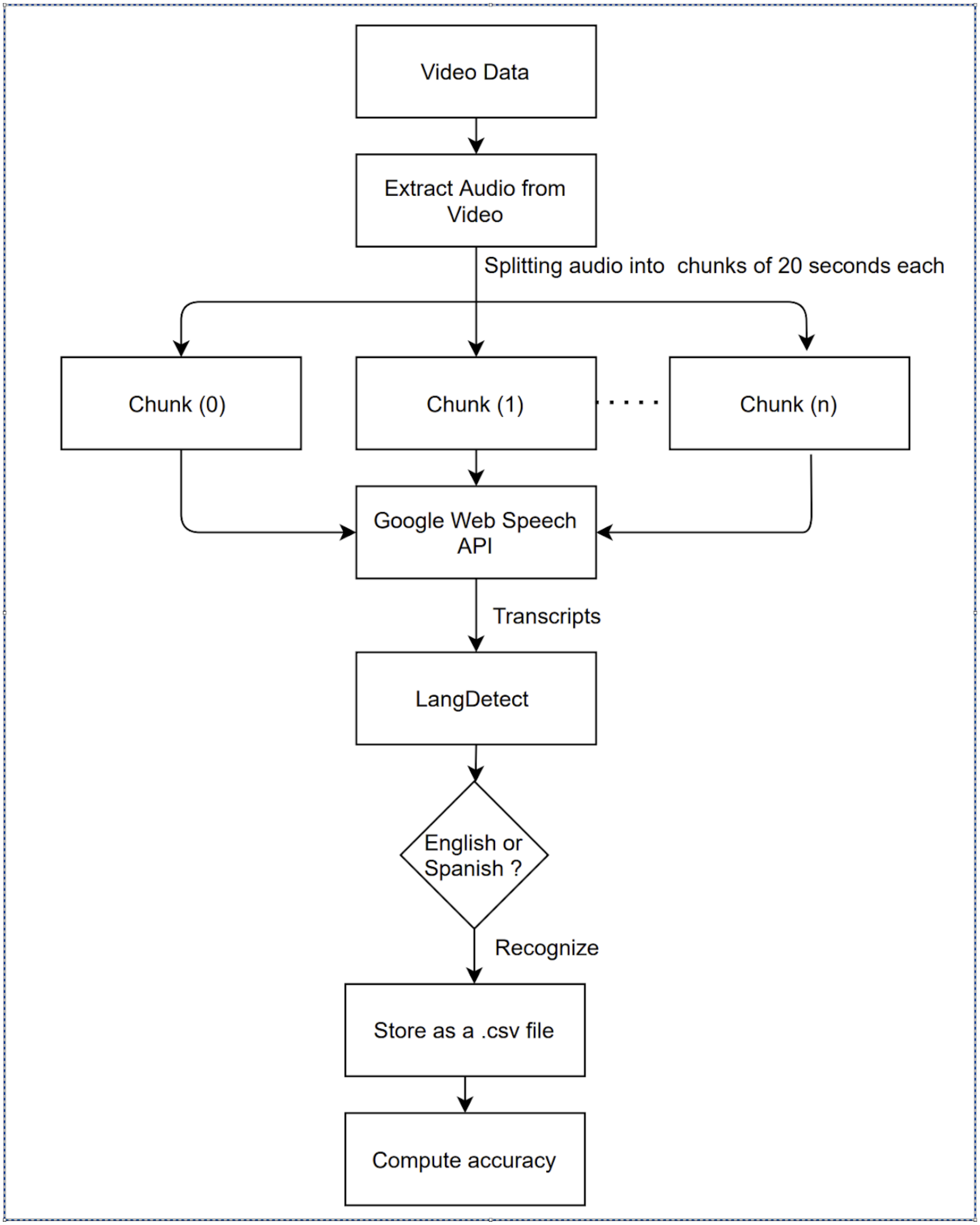

Figure 1. Design Diagram for Iteration 2

The software takes the video as the input. First, the audio is extracted. The audio is then split into files of 20 seconds each. Every 20 second file is fed to Google Web Speech API to get a transcript. Once the transcript is generated, it is fed to a language detector library, langdetect, to recognize English / Spanish. The predictions are stored in a csv file

The testing framework reads the above csv file stored and compares it with the files provided by the client. It shows the accuracy once done.

Integration:
We have independent functions for extracting the audio, splitting into a required number of audio files and ML model which makes API calls for Speech-To-Text and Text-To-Lang. These are integrated in which takes in the input video file and calls the functionality while maintaining formats; then encodes timestamp and language to save in a CSV format output.

Testing:
We have performed Unit tests in the functions. We performed integration testing in this iteration for the pipeline developed in python.

We are required to detect the language of instruction by the teacher as ‘Spanish’, ‘English’ or ‘None’; which are coded as 1, 2 and 5 respectively in the manual observation sheets provided by the client for each video. To test the accuracy of our code we have compared the actual language detected to the excel sheet values and evaluated the accuracy as the percentage of correctly detected chunks out of the total number of chunks.

Results:
The model has been tested on three different labelled xlsx files provided by the client. The process timings vary anywhere from 1 to 2 minutes. We have compared the output of the code against the language of instruction data and have observed the following accuracy percentages:

Filename	Percentage Accuracy
10112_R2.asf	92%
80222_R2.asf	80%
40631_R1.asf	35.13%

We observed that the code is biased towards the English language. It recognizes English really well but struggles in identifying Spanish.

For further research, we can refine the audio noise reduction models, use better speech recognition libraries and incorporate natural language processing toolkit models (NLTK library of

Python) and test against the data provided by the clients. We can take a step ahead and also explore tensorflow/scikit learn to understand the deep learning approaches to analyse audio data and the applications of deep learning language recognition.

Custom Grading:

1. Since there is no specific testing framework like Cucumber and Rspec, the testing is done in python only. Unit testing is performed within each module and integration testing is performed on the entire pipeline.
2. The testing performance for this project is evaluated based on the accuracy of the model that would be computed by comparing the extracted results with the existing videos.
3. Since this is a Machine Learning project that takes an input and extracts its features, it mainly works on data analysis and extrapolation and has no requirement of a front end user Interface.
4. Since it is a Proof of Concept application, the project has not been deployed to Heroku

Github Repository: https://github.com/rjagiasi/TBOP
Pivotal tracker account: https://www.pivotaltracker.com/n/projects/2495399