Defence innovation challenge 2020

Artificial Intelligence Track

Tasks

1. **Translation Task**
2. **Speaker Identification Task**

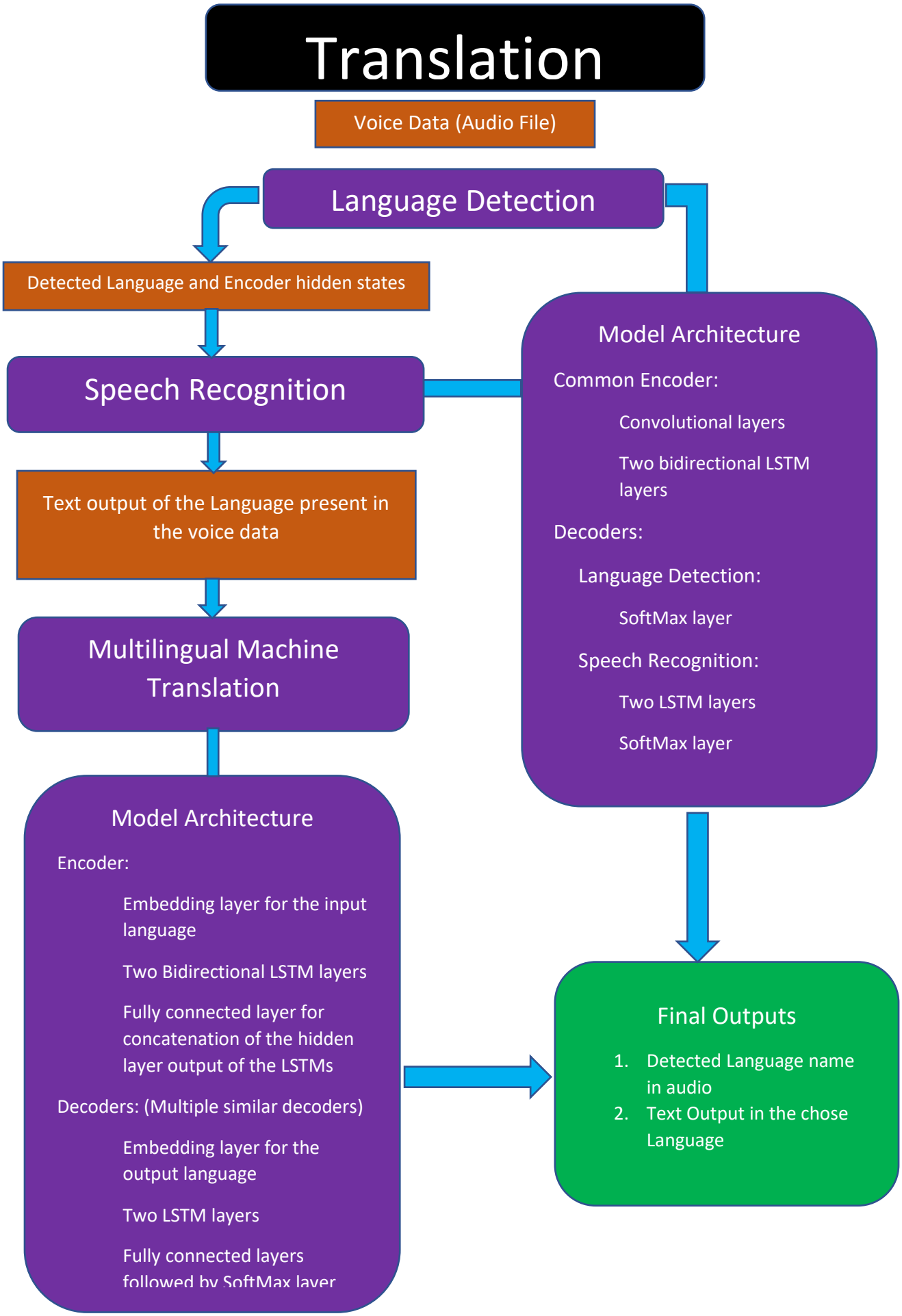
Contents

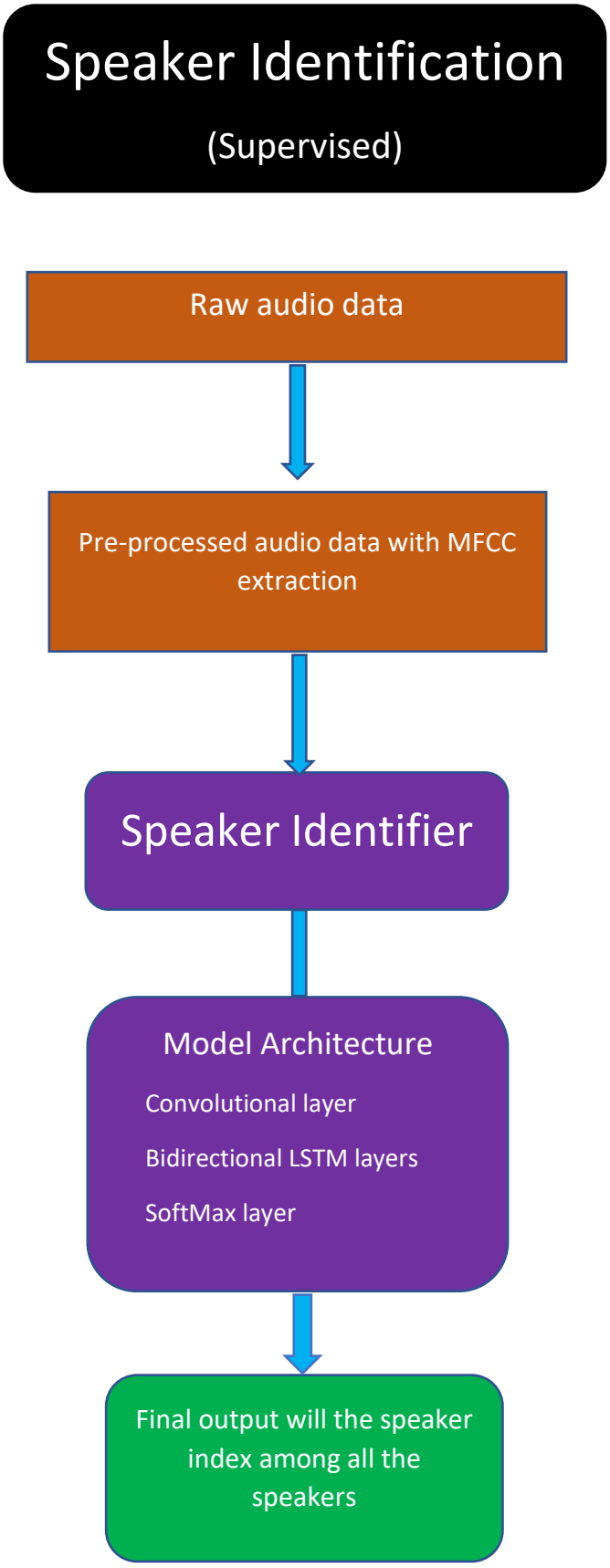
Team - Feynman

Yash Patel

Sachin Kumar

Flow Chart:


Translation
Voice Data (Audio File)
Language Detection
Detected Language and Encoder hidden states
Speech Recognition
Text output of the Language present in the voice data
Multilingual Machine Translation
Model Architecture
Encoder:
Embedding layer for the input language
Two Bidirectional LSTM layers
Fully connected layer for concatenation of the hidden layer output of the LSTMs
Decoders: (Multiple similar decoders)
Embedding layer for the output language
Two LSTM layers
Fully connected layers followed by SoftMax layer
Model Architecture
Common Encoder:
Convolutional layers
Two bidirectional LSTM layers
Decoders:
Language Detection:
SoftMax layer
Speech Recognition:
Two LSTM layers
SoftMax layer
Final Outputs
1. Detected Language name in audio
2. Text Output in the chose Language


Speaker Identification
(Supervised)
Raw audio data
Pre-processed audio data with MFCC extraction
Speaker Identifier
Model Architecture
Convolutional layer
Bidirectional LSTM layers
SoftMax layer
Final output will the speaker index among all the speakers

Description:

1. Translation task:

We propose that to translate the audio of an input language to a set of output languages, it is best to first classify the input language of the audio. This classification, acts as a input token for the further models to perform the translation more accurately. Hence, we have two models to perform the translation task:

1. **Language detection model –** It takes in the pre-processed audio converted into *frequency Spectrogram* format and classifies the input language from a set of output languages. It comprises of 2 Bidirectional-LSTM layers, stacked upon a convolution layer. The output of the final LSTM layer is fed to a SoftMax layer, which outputs a single token corresponding to the set of output languages.

Since we are dealing with a real-world scenario, we propose to also train the model, to classify the language as "None", if it is not confident of the input belonging to any of the set of output classes. This ensures that human interference is called at the right time and the information is not lost. Also, this prevents the Speech recognition model, to end up translating a language that it hasn't been trained for.

2. **Speech recognition and Translation model** – We divide the task of "speech recognition" and "translation to user chosen language" into two parts:

a. We train a **Speech recognition** model to convert the given raw audio to text format corresponding to the language in which it is spoken.

b. We train a **Machine translation** model, to translate the input language text to "text format" of the output language chosen by user.

This twostep process, allows us to use quality and large datasets of Audio to text in same language, such as those offered by Mozilla Common Voice initiative, and large corpuses of Machine translation (text to text) datasets. Bigger and quality datasets ensure that our model is able to train well to perform in real world scenarios.

Architecture of Speech recognition (Speech to text) model:

The architecture of the models works on the Encoder and Decoder concept. In the first model, the Encoder part is common for Speech recognizer and Language detector. The audio is being pre-processed in the form of *Spectrogram of frequencies* and fed to the Encoder. The output of the Encoder is given to a SoftMax layer with predicts the language present in the audio.
The encoder comprises of 3 Bi-LSTM layers stacked on top of a single convolution layer. While the decoder comprises of 3 LSTM layers stacked on top of each other, followed by a SoftMax layer at the output.

Architecture of Machine Translation (Text to Text) model:

For **Machine translation**, the model will be designed such that the same model can output multiple Languages. The model will be based on Google's natural Neural Machine translation idea [1]. In which first token to the Encoder part of the model will be a special token for a particular output language, which will be specified by user. In this model, there will be multiple decoders for different languages but the Encoder will be common for all the decoders. That will reduce the size of the decoder since it has to learn only for one language and this will be less computationally expensive at the time of testing. In addition to this, **Attention** will also be used in this model since attention has come out to be very useful in many Natural Language Processing tasks including Machine translation. The model will be trained on the "Europarl Parallel Corpus" dataset [2] for training **Teacher Forcing** will also be used.

2. Speaker Identification task:

The model proposed here is based on **Supervised learning**. It will be able to detect the person with audio input. First, it needs to get trained, for that "Mozilla's Common Voice" dataset will be used and based on the IDs of the speakers, classification will be done.
Such model with such architecture should be able to adapt to the new speaker quickly with very less data, in other words it should be able to do

Low-Shot learning. For that the model trained here can be used with **Transfer Learning** concept.
The model will be trained on “CSS10” for demonstration purpose.

In addition to these layers, all the models will contain some regularizing techniques e.g. DropOut, L2 regularization etc. for better accuracy of the models.

NOTE: The architecture specified above are little tentative in nature, in the sense that, since these are not tested fully by our team till now and have been selected on the basis of theoretical and experimental data from different articles and research papers and also from the past experience of our team members in Deep Learning field, therefore, it might happen that some of the hyperparameters like number of layers and positioning of layers in the architectures have to be changed based on testing later on.

References:

[1] Google’s Multilingual Neural Machine Translation Model

[2] Europarl Parallel Corpus

[3] Why Pytorch?

[4] Audio pre-processing for Speaker Identification

[5] Speaker Identification with Gaussian Mixture Models (GMMs)

Contact:

- GitHub links:
  - ComputerMaestro
  - sachin-101
- Emails:
  - Yash Patel - yp270200@gmail.com
  - Sachin Kumar - sachinkumar04428@gmail.com