Capstone project design

using a pre-trained multispeaker text-to-speech (TTS) model —> audio + text for each audio

Every Audio sample is paired with its transcribed speech

Audio : PyTorch audio — spectrogram, MFCC

Wav file —>

Text : word2vec

RNN ASR —model — saved model (reuse the saved model)

The trained ASR will be evaluated on unseen sentences for seen and from the multi-speaker TTS system

unseen speakers — different speakers

Speaker ID

Few speakers IDs —- used for recording

Id1, id2, id3 id4, id5

id1, id2, id3 —> voice sample + transcribed text used for training ASR model seen speaker + seen text)

id4, id5 —> voice sample but no text (to test performance of ASR) (unseen speaker unseen text) —> for testing

1. Completely unknown speaker(id6) not at all part of TTS training (for evaluation : unseen speaker unseen text)
2. Seen speaker but unseen text (id1, id2, id3 but text is new)

Which language :

English — stick to one language

Unseen text means prediction at word level

Language classification : phoneme (start and stop token)

Mappings from many word to one phoneme

Start with English and word-level processing

---

keep data loader ready by next week

RNN or special type of RNN like LSTM /GRU /bi-directional LSTM

Limit the number speakers for TTS and later scale

Initial local testing with small configuration and later scale

---

We need multi-speaker TTS data

NVIDIA TTS — upto 20 speaker IDs

Text corpus — start with 300-400 sentences

We will use text corpus for generating speech for TTS input