Self-supervised model learning pipeline to cater dysarthric speech

Dataset containing a mix of clear speech to dysarthric speech, which can be obtained from medical platforms. Speech audio can also include from interviews, podcasts, in single-speaker and multi-speaker conversations. Ensure dataset has at least sufficient samples, probably at least consisting of 10k hours.

Preprocessing measures to the audio can be done. First, convert all the audio data to sampling rate of 16kHz 16 bit PM through FFMPEG. Secondly, use a voice activity detection (VAD) model to filter and remove long silences that are more than 1 second. Next, try to segregate all audio samples to a certain maximum length with the detected long silences. Setting feature processing time window of 25ms and a 10ms window shift, convert all audio segments to 80-dim log-Mel features. Last but not least, use an audio event detection (AED) model to differentiate the speech part of the audio from the background noise with filters in real time like ignoring the utterance which doesn't have speech event, cropping utterances based on speech event to only include the speech part and random cropping for long utterances as data augmentation.

Setting up the pipeline for training, loss function to be used is the flatNCE, using AdamW optimizer and setting the training process to be of 2 million steps. Metric to measure training and validation processes is word error rate. The learning rate is set at a maximum of 1e-3 with the first 10% of steps as warm-up to reach the maximum learning rate and the rest of 90% with a linear decay to reach 5e-6.

The model pipeline undergone is a non-streaming SSL pretraining. What happens is first, log-Mel features are masked as well as initial time steps sampled randomly to be masked with a probability and then mask the subsequent 10 time steps. After which, the masked features are to the encoder which can be a 6-layers bi-directional LSTM with 600 hidden dimensions, as well as a 20 dimensional linear projection nlayer and L2 normalized to retrieve masked context vectors. Running in parallel, log-Mel features are also passed to another 20 dimensional linear projection layer and L2 normalized to retrieve target vectors. The contrastive loss between the masked positions of context vectors and target vectors is optimized.

How to do continuous learning for the SSL model

After training the SSL model, finetuning can be done by first freezing the LC-BLSTM layers initialized from the model and the model's weights, only training for the model's head which is the linear projection layer initialized from scratch. After training the linear projection layer to convergence, unfreeze the entire model and finetune the whole model. During finetuning, different learning rates can be toggled like a larger learning rate for faster convergence of model then change to a smaller learning rate for stable convergence of model afterwards.