Essay question – please read https://arxiv.org/pdf/2205.08598.pdf and propose a model self-supervised learning pipeline to cater dysarthric speech and describe how you would do continuous learning in 500 words. Your answer can be saved as essay-ssl.pdf under the main repository.

Summary of "Deploying self-supervised learning in the wild for hybrid automatic speech recognition"

The paper discussed about self-supervised learning (SSL) methods in automatic speech recognition (ASR). In their paper, they noted that most prior SSL-ASR research uses clean, curated datasets, and from their end, they wish to work on uncurated, noisy, real-world audio data for training and deploying ASR models.

Hence, they proposed a full SSL pipeline, from data preprocessing to pretraining, fine-tuning, and deployment, targeting streaming hybrid ASR that is suitable for production systems. To do so, they utilized a Xception-based audio event detection (AED) model to remove non-speech segments like music, alarm, crowd noise, which improved data quality, lower noise, and reduces Word Error Rate (WER) significantly.

Next, they introduced Lfb2vec, a self-supervised model that uses masked log-Mel features with contrastive learning, trained with BiLSTMs, optimized for hybrid ASR. Additionally, they tested various contrastive loss (InfoNCE vs flatNCE) and deduced that flatNCE is more stable with smaller batch size and provides better WER.

Various pretraining strategies are tested: in- vs out-domain, mono- vs multi-lingual, supervised vs self-supervised, multi-head vs single-head multilingual SSL. Ultimately, an in-domain SSL with AED improves training efficiency and model performance while flatNCE loss function with AdamW optimizer are more stable for longer training. Their approach proves to work with massive, multilingual, and noisy audio sources, which are good for real-world applications.

Dysarthric speech

Going back to the problem statement, we wish to propose a model self-supervised learning pipeline to cater dysarthric speech. Dysarthric speech refers to speech that is impaired due to weakened or uncoordinated muscles involved in speaking. It's a motor speech disorder often caused by neurological conditions not limited to stroke, traumatic brain injury (TBI), Parkinson's disease, amyotrophic lateral sclerosis (ALS), and cerebral palsy.

The characteristics of dysarthric speech are not limited to slurred or mumbled speech, irregular pitch, rhythm, or volume, imprecise articulation, slow or rapid rate of speech, or

monotone voice. These characteristics make dysarthric speech challenging for ASR systems, especially those trained only on standard or “fluent” speech. Traditional ASR models often perform poorly on dysarthric speech due to its high inter-speaker variability and non-standard acoustic patterns.

Connection between paper and dysarthric speech SSL

The work performed in the paper is well suited for the use-case of dysarthric speech for several reasons. First, as it is self-supervised, we do not require transcription to train the model. Transcription is often the costliest portion of an audio dataset, and removing the need for transcription makes the training more flexible. Secondly, we can still fine-tune the model with some labelled data to make it more adaptive to diverse and atypical speech patterns. Finally, if we have patient-specific data, we can even enable personalized models through continued learning from user-specific data.

Possible implementations

To test out the concepts in terms of a proof-of-concept (POC), we can start with a self-supervised learning model like Lfb2vec or Wav2Vec2. Then, we can obtain some mildly impaired speech to pretrain the models, to allow them to generalize for more challenging tasks. Next, we can find some open sourced or internal labelled dysarthric data and fine-tune the models using the loss functions stated in the paper and apply AED preprocessing to remove silence and noise to make the speech more audible.

When the model is deployed and new audio data are available, we may apply two stage fine-tuning: update only the speaker embedding or projection head or periodically update full model using selective rehearsal (reply buffer of old samples). We can use LC-BiLSTM architecture for low latency real time ASR. We can also use the ASR model to generate transcript for unlabelled data, manually verify them, and use them for fine-tuning and retraining. For incremental updates, we use the flatNCE loss as advised in the paper, which improves representation learning.

Finally, we continuously track model confidence and error rates; if uncertainty raises, flag examples for manual checking or retraining.