Proposal for Model Self-Supervised Learning (SSL) Pipeline for Dysarthric Speech

Assumption: We are looking to solve ASR English Dysarthric Speech

Model Proposal

a) Data Collection & Pre-processing

- **In-domain audio sourcing**: Gather hours of unlabeled dysarthric recordings covering diverse speaking rates, severities, and environmental contexts (noisy environments, volume etc).
- **Out-domain audio sourcing**: Gather out-domain (regular speech) recordings (in other languages or accents to enhance shared phonetic learning). This will be to simulate multilingual data in the original paper. If necessary, we can employ data augmentation to try and simulate dysarthric speech to increase the amount of dysarthric recordings for pre-training.
- **Standardization**: Convert audio to 16 kHz 16 bit PCM through FFMPEG
- **Voice Activity Detection (VAD)**: Remove silences > 1 sec and split into ≤20 sec segments.
- **Feature Extraction**: Convert audio segments to 80-dim log-Mel features, with feature processing time window as 25-ms with 10-ms window shift
- **Audio Event Detection (AED)**: Use an Xception classification model (speech/music/noise detector). Employ both AED filters:
  - **Speech-filter**: discard non-speech segments.
  - **Speech-crop**: isolate spoken content.
- **Augmentations**:
  - **Rand-crop**: randomly trim longer segments .

b) Pre-Training

- **Masking**: Randomly mask log-Mel features time spans (~0.065 probability × 10 frames), replicating Lfb2vec.

- The masked features are then passed into two pipelines
    - **Path 1**
        - **Encoder**: Use 6‑layer Bi-directional LSTM with 600 hidden units (used by the paper)
        - 20-dimensional linear projection
        - L2 Normalisation
    - **Path 2**
        - 20-dimensional linear projection
        - L2 Normalisation

- **Contrastive Loss function**: Use flatNCE instead of InfoNCE to reduce bias and batch‑size sensitivity.

- **Optimizer & Scheduling**: AdamW with warm‑up (~10%) + linear decay over millions of steps

- **Multi‑head Lfb2vec**: Share encoder across speech types (dysarthric vs. healthy) and severity levels; each head projects into its domain-specific feature space.

c) Fine-Tuning

- On labeled dysarthric speech, fine‑tune a hybrid streaming ASR using 6-layer LC-BLSTM followed by linear projection layer as the senone-based Acoustic Model (AM)
    - AM model is trained with 20 frames chunk length, 20 frames look ahead length, and cross entropy loss function on target language

- Use two‑stage fine‑tuning:

    Train linear layer while freezing encoder
    - Unfreeze encoder layers

- Decoding Stage: Use a 5‑gram LM adapted to dysarthric word patterns.

Continuous Learning

a) Unlabeled Data Ingestion

- Continuously feed recorded user speech into the pre-processing + SSL pipeline.
- Re-run AED + data cleaning to create new unlabeled examples for adaptation.

b) Periodic SSL pre-training

- Every N hours or epochs, retrain the SSL model with freshly ingested unlabeled in-domain speech,

c) Continual Fine-tuning

- Weekly or monthly, fine-tune acoustic model using a mix of:
    - Pseudo-labels (using existing ASR model),
    - Historical + new labeled dysarthric data (budget dependent)

d) Evaluation & Feedback Loop

- Deploy evaluation protocols:
    - Track WER on labeled test sets,
- If performance drops beyond threshold:
    - Increase data ingestion,
    - Reset to last stable checkpoint,
    - Adjust learning schedules dynamically.

Summary

We enhance data pre-processing (AED, augmentations) and use flatNCE loss with stable optimizers to train dysarthric-specific representations. Multi-head architecture learns from both healthy and dysarthric speech. Continuous learning integrates fresh unlabeled and labeled data, updating both SSL encoders and ASR models. We make use of evaluation metrics & rollback to ensure robust, personalized performance over time.