xDATA Technical Test 2023

Question 6

Self-Supervised Learning for ASR: Dysarthric Speech

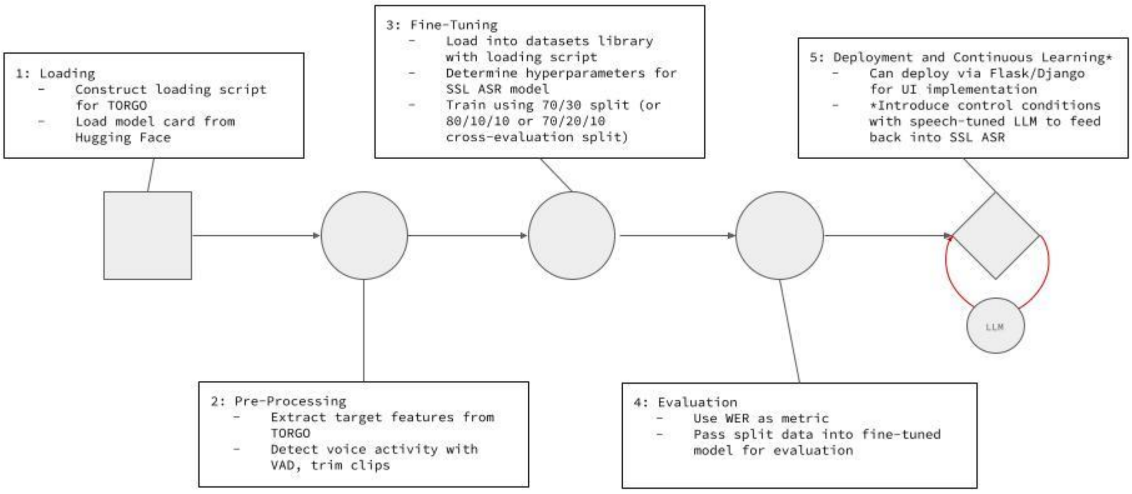
Clarence Teo Kai Xuan (clarenceteokx@gmail.com)

Introduction

Dysarthria can be classified as the condition in which muscles involved in the production of speech have atrophied. As such, dysarthric speech has unique acoustic and supralinguistic characteristics that distinguishes it from modal speech. This poses an unique opportunity in automatic speech recognition (ASR), pertinently in self-supervised learning (SSL) models.

Methodology

The SSL-ASR pipeline (Fig. 1) for dysarthric speech is influenced by [1], with the difference made in terms of the dataset selection and application of continuous learning.



1: Loading
- Construct loading script for TORGO
- Load model card from Hugging Face
2: Pre-Processing
- Extract target features from TORGO
- Detect voice activity with VAD, trim clips
3: Fine-Tuning
- Load into datasets library with loading script
- Determine hyperparameters for SSL ASR model
- Train using 70/30 split (or 80/10/10 or 70/20/10 cross-evaluation split)
4: Evaluation
- Use WER as metric
- Pass split data into fine-tuned model for evaluation
5: Deployment and Continuous Learning*
- Can deploy via Flask/Django for UI implementation
- *Introduce control conditions with speech-tuned LLM to feed back into SSL ASR
LLM

Figure 1: *SSL-ASR Model Training Pipeline*

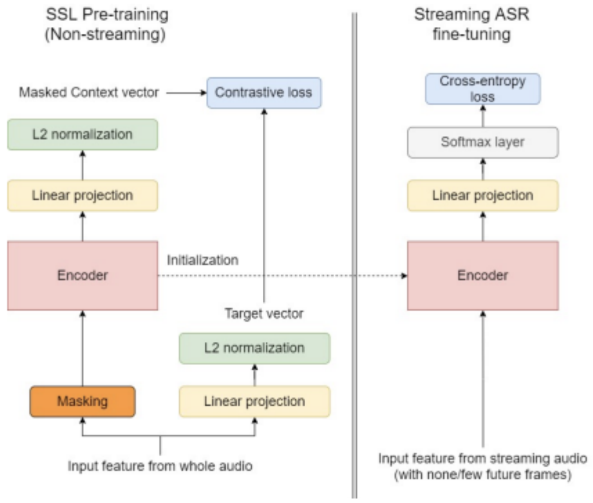


Figure 2: *SSL-ASR model architecture [1]*

First, a loading script will be written and run on the TORGO dataset [3]. The specified model will also be loaded via its model card. Second, preprocessing steps will be conducted (i.e., downsampling to 16000 Hz 16-bit; voice activity detection (VAD) to remove silences; log-Mel audio decomposition). Third, the model (Fig. 2) will be fine-tuned on these proposed hyperparameters:

- warmup_steps: 10% of total steps taken for fine-tuning
- evaluation: 500 steps
- metric: 'wer'
- optimiser: AdamW (weight decay after controlled parameter-stepped size)
- learning_rate: 1e-04
- train/eval_batch_size: dependent on compute memory (default: 8/16, 1-4 for local non-commercial GPUs)
- mixed precision method: fp16 (saves memory)

Fourth, for the evaluation of fine-tuning steps, word error rate (WER) will be used as a metric for ASR transcription systems. As a benchmark for low-resource speech, 30-40%WER would be expected. Finally, the model could be deployed on Flask or Django as an inference API (also possible through HuggingFace), with control conditions introduced by an instruction, speech-tuned LLM (e.g., [4]) to select audio samples for the model's iteration of continuous learning.

As a consideration, fine-tuning would pose a problem due to the relatively small TORGO dataset. As such, it would make continuous learning (Fig. 1, 2) an important aspect of modelling ASR for dysarthric speech.

Continuous Learning

On the front of continuous learning, it describes the integration of up-to-date data with the models created to keep the outputs relevant to the projected use case (and users). This can be done in three ways: (a) incremental training, where new data is continuously fed into the model and fine-tuned, (b) batch training, where training takes place after a certain threshold of new data is generated, and (c) retraining, where ground-zero training takes place after the data threshold is reached [2].

The model may undergo a few changes to the architecture in order to support continuous learning, namely:

(1) Continuous monitoring and recording: Subject to model production and deployment, an additional component could be included where user input audio and metadata can be stored (with permission) for model fine-tuning purposes. An automated back-end feedback loop (e.g., monthly/ bi-monthly basis) could be implemented where any of the three continuous learning techniques during a maintenance period. The new model could then be deployed for use after maintenance.

(2) Inclusion of Audio Event Detection (AED) in pre-processing pipeline: this is to account for potential noise in the new data points, which could be recorded in a multitude of environments. The distinguished audio and environment noise could then be separately used to feed into the SSL-ASR and denoising models, respectively.

In the proposed pipeline above, the first option was selected as a potential option for continuous training. Individuals utilising the inference API could potentially exhibit dysarthric speech and hence, be suitable data candidates for the continuous learning capacity of the model.

References

[1] M. Karimi, C. Liu, K. Kumatani, Y. Qian, T. Wu, and J. Wu, "Deploying self-supervised learning in the wild for hybrid automatic speech recognition," in *ICASSP 2022*, 2022.

[2] N. Sanjay, "Continuous Training of ML models", 2022, from https://medium.com/@nagasanjayvijayan/continuous-training-of-ml-models-7d8acaf44dda.

[3] F. Rudzicz, A.K. Namasivayam, & T. Wolff, The TORGO database of acoustic and articulatory speech from speakers with dysarthria. *Lang Resources & Evaluation* 46, 523–541 (2012). https://doi.org/10.1007/s10579-011-9145-0

[4] Fathullah, Y., Wu, C., Lakomkin, E., Jia, J., Shangguan, Y., Li, K., Guo, J., Xiong, W., Mahadeokar, J., Kalinli, O. and Fuegen, C., 2023. Prompting large language models with speech recognition abilities. *arXiv preprint arXiv:2307.11795*.