Chapter 3

METHODS

To investigate whether training a model on L2 speech can improve its ability to recognize dysarthric speech, this study finetunes Wav2vec2 (Baevski et al., 2020) on different datasets that either include or excludes L2 speech data. This section of the thesis describes the tools, datasets, preprocessing pipelines, and evaluation methods used to conduct the study's experiments. The precise experimental setup is described in Chapter 4.

3.1 Datasets

The study uses two dysarthric speech datasets, TORGO and UA-Speech, and L2-Arctic, an L2 speech dataset. It uses two dysarthric speech datasets for several reasons: (1) to better balance the number of dataset samples containing L1, dysarthric, and L2 speech, (2) to enable the model to learn more robust representations of dysarthric speech, and (3) to evaluate how well different models can generalize the information they learn from both datasets. Table 3.1 shows a breakdown of samples found in each dataset.

Table 3.1: Number of audio samples within each dataset

Dataset	L2Arctic	UA-Speech	TORGO
Dysarthric	-	11,437	30,94
Control	-	9,945	5,900
Total	26,877	21,382	8994

The Universal Access, or UA-Speech, dataset consists of English speech from 15 people with cerebral palsy (4 female and 11 male) and 13 people without dysarthria (4 female and 11 male) (Kim et al., 2008). 11 participants had a diagnosis of spastic dysarthria. The remaining 5 were diagnosed with athenoid dysarthria or a mix of dysarthria subtypes. One speaker, M06, is not included in the dataset used in this study because he did not consent to his data being redistributed. The dataset consists of wav files of single-word utterances from a microphone array of 8 microphones, sampled at 16kHz. The `noisereduce` algorithm (Sainburg, 2019) was used to remove noise from the recordings.[1] The recordings are divided into three blocks. Participants produced utterances of the same 155 words for each block. They also produced speech for 100 words that differed across blocks, for a total of 765 utterances and 455 unique words (Kim et al., 2008). In this study, recordings of the same prompts were included in the dataset and are treated as separate data points.

The TORGO dataset consists of English speech from 7 people (4 male, 3 female) with dysarthria and 7 people (4 male, 3 female) control participants (Rudzicz et al., 2011). All participants with dysarthria had CP, and one had both CP and ALS. The dataset consists of wav files of nonce utterances and single-word or multiple-word utterances taken from the TIMIT corpus. The recordings are saved as wav files and sampled at 16kHz. Recordings were collected across three sessions. The speakers were tasked with reading as many prompts as they could within each time-limited session. So, some speakers have multiple recordings of the same prompts, while others did not complete every prompt provided.

The TORGO and UA-Speech datasets categorize the intelligibility of dysarthric speech differently. For the UA-Speech data, five listeners with no background in language disorders or phonetic transcription provided transcriptions for each speech recording. Speakers were placed into very low, low, medium, and high intelligibility categories based on the accuracy scores of human raters. The TORGO dataset provides Frenchay assessment scores for each speaker with dysarthria, obtained by a speech-language pathologist (Rudzicz et al., 2011).

---

[1] The database was updated to include recordings with this preprocessing step in 2020.

The Frenchay assessment assesses people's ability to move their articulators by asking them to perform tasks like talking or swallowing water. The assessment includes an intelligibility category consisting of three tests for speech interpretability (Enderby, 1983 in Rudzicz et al., 2011).

The L2-Arctic dataset includes recordings of spoken English by 24 non-native English speakers, distributed evenly by gender and L1 (Zhao et al., 2018). The speakers' L1s were Arabic, Mandarin, Spanish, Vietnamese, Korean, and Hindi. Speech recordings for each language were obtained from 2 male and 2 female speakers for each L1. The dataset consists of wav files of short, prompted sentences sampled at 44.1kHz. For this study, the L2-Arctic data was converted to a sample rate of 16Hz.

3.2 Finetuning Wav2vec2

The models for this study were created by finetuning the Wav2vec2 base model on the UA-Speech, TORGO, and L2-Arctic datasets. The Wav2vec2 model is trained on 960 hours of the Librispeech dataset Baevski et al., 2020, which consists of English recordings of audiobooks (Panayotov et al., 2015). Most of the dataset contains L1 English speech. Examining the dataset's metadata, about 2-3% of the Librispeech data is drawn from audio samples spoken by non-native English speakers.[2]

The Wav2vec2 model was downloaded and modified using the `transformers` package (Wolf et al., 2020) for Python. The experiments use the 960-hour checkpoint of the model, which finetuned the unsupervised model on 960h hours of audio from the Librispeech dataset. The study uses single linear layers as decoders for all experiments and paradigms. These decoders map the outputs of the Wav2vec2 model to a set of English characters.

Pasad et al. (2021) found that Wav2vec2 encodes less linguistic content in the last few layers and that these layers change the most during finetuning, leading them to suggest

[2] This figure may be inaccurate since the creators of the dataset state that its annotations are unreliable. Regardless, the percentage of L2 speech in the dataset is likely small.

reinitializing these layers before finetuning the model. So, before training, the weights of the last 2 layers of the transformer decoder are reinitialized.[3]

While conducting the study, several models using different configurations were generated for hyperparameter tuning and evaluating various model configurations. These models were assessed using the validation set. Information on hyperparameter selection and early exploratory experiments can be found in Appendix B, one of which serves as a sensitivity analysis for the study.

3.2.1 CTC Loss

Wav2vec2 models are finetuned by calculating Connectionist Temporal Classification (CTC) loss. CTC loss maximizes the probability that the model input corresponds to its label. Contextual representations are obtained from the final hidden layer of the Wav2vec2 transformer. These representations can be transformed into grapheme probabilities using the softmax function. The CTC loss algorithm compares these fixed-length sequences of grapheme probabilities to labeled text of variable length. The output sequence is the same length as the contextual representation generated by the model (the quantized units in Wav2vec2); a longer output sequence is compared to shorter labels by reducing sequences of identical graphemes into a single grapheme and using a special blank grapheme label to mark a sequence of repeated graphemes. The alignment reduction leads to a many-to-one mapping between alignments and labels.

Let B denote the function that reduces alignments, and its inverse B^{-1} denote a mapping between a label Y and a set of corresponding alignments. Assuming conditional independence, the total probability that a sequence of grapheme probabilities, X, corresponds to a label, Y, is:

$$P(Y|X) = \sum_{A \in B^{-1}(Y)} \prod_{t=0}^{N} P(a_t|X) \tag{3.1}$$

[3] The weights are reinitialized by sampling from a uniform distribution per Huggingface's implementation of Wav2vec2 on GitHub.

where $A = [a_1, ..., a_n]$ is a sequence in the set of sequences that maps to the label Y. The value $P(Y|X)$ can be efficiently calculated using a modified version of the beam search algorithm (Jurafsky and Martin, 2022; Hannun, 2017).

The model then minimizes the negative log-likelihood of the input mapping to the correct label. For a set of labels L and a set of inputs I, the loss $\mathcal{L}$ is calculated as follows:

$$\mathcal{L} = -\sum_{Y \in L,\ X \in I} \log P(Y|X) \tag{3.2}$$

Because the dysarthric datasets largely contain single-word utterances, while the L2-Arctic dataset contains multi-word sentences, the CTC loss for samples taken from the L2-Arctic dataset tends to be larger than for samples from the dysarthric datasets. To avoid ascribing higher losses to samples from the L2-Arctic dataset, all CTC losses were divided by the number of characters in the true labels, effectively normalizing the loss values across datasets. The loss for a batch is calculated by averaging the losses for all batch inputs. So, for a batch of inputs $B = \{x_1, ..., x_k\}$ with labels $\{y_1, ..., y_k\}$ of lengths $\{T_1, ..., T_k\}$ the loss is calculated as follows:

$$\mathcal{L} = -\frac{1}{|B|} \sum_{i=1}^{k} \frac{\log P(y_k|x_k)}{T_k} \tag{3.3}$$

where $P(y_k|x_k)$ is calculated using equation 3.2.1 and $|B|$ is the number of items in the batch. Examining CTC losses of a subsample of the data during the first epoch of finetuning found that dividing the CTC losses by sequence lengths T_k resulted in dysarthric speech receiving higher loss values (L2 mean loss: 2.18, dysarthric mean loss: 5.52). Without normalizing, we observed the opposite pattern (L2 mean loss: 44.99, dysarthric mean loss: 28.58), which was not desirable given the lower baseline WERs of the L2 data.

3.3 Evaluation

The current study evaluates model performance using word error rate (WER) and character error rate (CER). WER is based on the minimum edit distance (or Levenshtein distance) algorithm, but it calculates substitutions, deletions, and insertions at the word level instead of the character level (Jurafsky and Martin, 2022). After computing the minimum edit distance between the predicted sentences and their corresponding labels, WER is computed as follows:

$$WER = \frac{S + I + D}{N = H + S + D} \tag{3.4}$$

where S, I, and D are the count of substituted, inserted, and deleted words respectively, N is the total number of words in the label, and H is the number of correct words (Morris et al., 2004).

Because many of the utterances in the dysarthric speech datasets are single words, it is helpful to consider character error rate (CER). CER is calculated using equation (3.3), but the counts for I, S, D, and H are obtained at the character level. However, because English orthography has little correspondence with the phonetic realizations of its represented words, CER may not be a good measure of model performance or of partial correctness.

3.3.1 Issues with WER and CER

Although WER and CER are standard metrics in ASR studies, including dysarthric speech recognition studies, there are significant drawbacks to their use in this context. As noted earlier, there is little correspondence between English orthography and its phonetic realization. Training a model to generate English orthography implicitly trains it on English spelling conventions, obscuring our understanding of how the model processes phonetic information. The metrics provide little insight into how a model internally represents phone segments since the metrics don't account for phonetic features associated with each segment. For

example, the phones [p] and [b] in speech only differ in whether they are voiced (or [+voice] in distinctive feature theory), but CER and WER consider this substitution to be equivalent to replacing [p] with [i], even though the latter differs in voicing place of articulation, and manner of articulation among other differences. It also fails to account for homophony. For instance, two homophonous words (such as "male" and "mail") would be given a WER score of 1 despite being phonemically identical.

Despite their limitations, this study uses WER and CER because no pretrained monolingual Wav2vec2 models are trained on phonemic transcription. Creating a Wav2vec2 model trained on phonemic transcription was not considered since it would require resources not commensurate with the study's scope. While multilingual Wav2vec2 models are trained on phonemic transcription, including other languages would present a confounder for this study.

3.3.2 Matched-Pair Sentence Segment Word Error

The Matched-Pair Sentence Segment Word Error (MAPSSWE) is a parametric statistical test that evaluates whether model outputs are significantly different (Jurafsky and Martin, 2022; Gillick and Cox, 1989). The test divides sequences of words into segments composed of one or more words and calculates a score, W, which is the difference in the number of errors each model makes within a segment. The advantage of segmenting the data is that it allows us to assume that errors across segments are independent if the data is segmented at a natural stopping point such as a pause (Gillick and Cox, 1989). If systems have similar WER, the mean difference in errors would be 0, or $\hat{\mu}_z = \frac{1}{n}\sum_{i=0}^{n} Z_i = 0$. In other words, the null hypothesis, H_0, is that the two WERs are not significantly different.

With a sufficiently large n, the distribution of errors W, should be approximately normal, allowing us to calculate:

$$W = \frac{\hat{\mu}_z}{\sigma_z/\sqrt{n}} \qquad \text{where, } \sigma_z^2 = \frac{1}{n-1}\sum(Z_i - \hat{\mu}_z)^2 \tag{3.5}$$

We can then calculate, $P(Z > |w|)$ where w is the realized value of W and $Z \sim \mathcal{N}(0, 1)$. If $P(Z > |w|) \geq \alpha$, for a significance level α (set to 0.05 in this study), we reject the null hypothesis (Gillick and Cox, 1989). We use the two-tailed version of the test to compare models trained with L2 and dysarthric speech to those trained on solely dysarthric speech since we are also interested in whether L2 speech significantly worsens model performance. We used the National Institute of Standards and Technology's SCTK software to segment text and calculate MAPSSWE scores (*SCTK* 2021).[4] MAPSSWE test were only performed on the dysarthric speech data—L2 and control data was removed before conducting the tests.

It is important to note that the MAPSSWE test assumes that the errors are normally distributed (Gillick and Cox, 1989). However, because most of the dataset consists of single-word utterances, most of the errors fall take on values of -1, 0, or 1. The limited range of errors and use of discrete values make it difficult to evaluate whether W will be approximately normal. For that reason, the test's outcomes should not be regarded as definitive evidence of performance differences across models. Nevertheless, we report the test's results since it is a helpful tool for comparing model performances, especially because it can be challenging to determine whether WER and CER values are substantially different.

3.4 Training paradigms

Two training paradigms are compared in this study: finetuning and multitask learning.

3.4.1 Finetuning

The finetuning paradigm involves training a pretrained model on additional data to transfer knowledge from its original domain to new domains or downstream tasks (Jurafsky and Martin, 2022). In this study, the finetuning paradigm is implemented with two datasets: one containing both the dysarthric speech datasets and L2-Arctic and another containing only the dysarthric speech datasets. The latter serves as a control group for the experiments.

[4] The software can be downloaded from Github: https://github.com/usnistgov/SCTK.

Wav2vec2 is then finetuned on these two dataset configurations. The weights of the last two layers of the transformer are reinitialized before training, as suggested by Pasad et al. (2021).

3.4.2 Multitask

Multitask learning also aims to facilitate knowledge transfer, but it accomplishes this goal by training a model on two tasks simultaneously (Zhang and Yang, 2017). This study's multitask training procedure adapts the finetuning paradigm with a different model architecture. It branches the last two layers of Wav2vec2's transformer-based encoder into two separate branches. In other words, inputs for each task are passed into separate copies of the final 2 layers. The motivation for separating the final layers for each task was to allow the model to encode contextual information specific to each domain. Here too, the weights of these last two layers are reinitialized at the start of training, as suggested by Pasad et al. (2021).

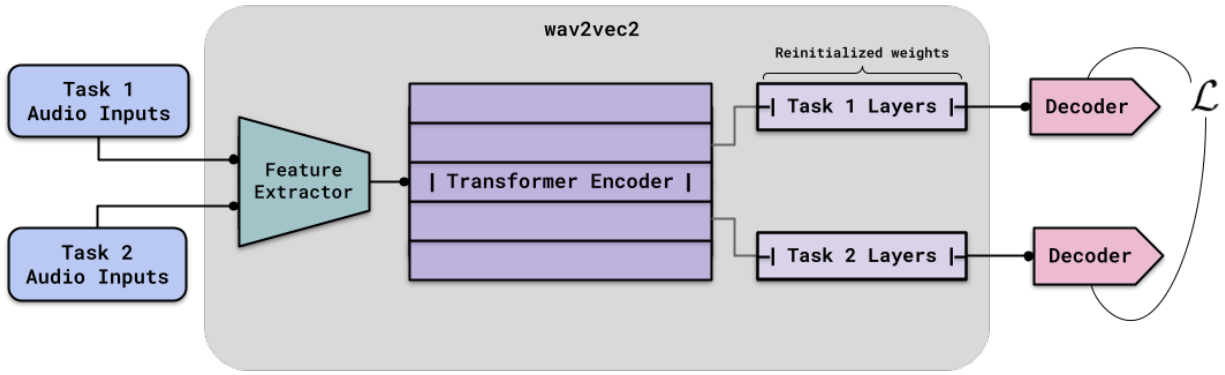

Figure 3.1: A diagram of the multitask model. The transformer encoder branches off at the last two layers before passing its outputs to separate decoders.

Tasks indices are provided as inputs to the multitask models to identify the branch to which the model needs to forward inputs. The branches then provide outputs to two separate decoders consisting of a single linear layer. The multitask models are constructed by duplicating the weights from the linear layer that obtains grapheme probabilities from Wav2vec2's encodings. A diagram of the multitask configuration can be found in Figure 3.1.

Initially, we considered passing the control data from the dysarthric speech datasets to a separate branch. Earlier experiments found that passing the control and dysarthric data to a single branch proved more effective, leading us to pass all data from dysarthric speech datasets to the same branch.

3.5 Chapter summary

This chapter describes the datasets, training paradigms, and evaluation measures used in the study. The chapter describes the training protocols to finetune to models and the architecture of the multitask model. It also describes the corpora used by the study: the UA-Speech, TORGO, and L2-Arctic corpora, and the resulting balance of native English non-dysarthric, native English dysarthric, and L2 speech in the dataset. The models are trained on this data using the CTC objective, normalized by label lengths. Despite their theoretical shortcomings, models are then evaluated using word error and character error rates. Performance is compared across models using the Matched-Pair Sentence Segment Word Error statistic.