THE UNIVERSITY OF GRONINGEN

MASTER THESIS

Dutch Dysarthric Speech Recognition: Applying Self-Supervised Learning to Overcome the Data Scarcity Issue

Author:
Tatsunari MATSUSHIMA

Supervisors:
Dr. Shekhar NAYAK
Dr. Matt COLER

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Voice Technology
Campus Fryslân

July 15, 2022

THE UNIVERSITY OF GRONINGEN

Abstract

Voice Technology
Campus Fryslân

Master of Science

Dutch Dysarthric Speech Recognition: Applying Self-Supervised Learning to Overcome the Data Scarcity Issue

by Tatsunari MATSUSHIMA

Automatic speech recognition (ASR) has been successfully used for many applications. However, the development of ASR for dysarthric speech, a common pathological disordered speech, has been hindered due to the lack of training data. Since supervised learning is a data-hungry approach that demands expensive manual annotations, it is not optimal to develop ASR for dysarthric speech. Motivated by successful applications of self-supervised learning (SSL) in ASR for low-resource languages, which have a similar condition of the data limitation, the research applies SSL for Dutch dysarthric speech recognition for the first time. The state-of-the-art model, wav2vec 2.0, and XLSR-53, a cross-lingual model of wav2vec 2.0, are used for benchmarking. The results show that the SSL models achieved poorer performance than the supervised DNN-HMM model. However, the author observed the SSL model's superiority in the generalization ability among different severity groups and patients. Since the dysarthric speech features significantly differ depending on the severity, type of disorder, and speaker characteristics, it is assumed that the generalization ability potentially degrades the SSL model's performance. Hence, the research further develops the speaker-dependent ASR for dysarthric speech. The results show that only roughly 10 minutes of re-fine-tuning with the target speaker's utterances significantly improves the models' performance, achieving 10.79 WER at the highest. It demonstrates how speaker-dependent SSL can eliminate the data limitation constraint in developing dysarthric speech recognition. This is an imperative milestone to developing a working-level Dutch dysarthric speech recognition. The author summarizes the outcome as an SSL training strategy framework for dysarthric speech recognition to catalyze future research.

Contents

Chapter 1

Introduction

Automatic speech recognition (ASR) has been successfully applied to many commercial products. These applications have brought benefits to humans. Automatic transcription apps can reduce the efforts of note-taking and making captions. Voice assistance enables voice-driven control over digital devices, which is particularly useful when users' hands are occupied, as when driving, operating machinery, or similar. It is also useful in cases where a speaker has a disability, for example. Although the success of ASR is remarkable, where the accuracy benchmarks of ASR are almost above 95% [1], [2], recognizing pathological speech is still very challenging as the speech often lacks intelligibility [3]. Dysarthria, a motor speech disorder that causes interferences in respiration, laryngeal function, airflow direction, and articulation resulting in low speech intelligibility [4], is one common pathological disorder of speech. Commercially available cloud-based ASR applications, namely IBM Watson Speech-to-Text, Google Cloud Speech, and Microsoft Azure Bing Speech, were investigated for pathological speech recognition [5]. The limitations of these applications were reported with 80-90% word error rate (WER). This result indicates that people with communication disorders can not fully obtain the benefits of ASR applications. Therefore, the utility of an ASR system that functions not only with healthy speech, but also with pathological speech is evident. Such an advancement would represent both scientific as well as social progress.

The main challenge in the development of speech recognition for pathological speech is the lack of available data [3], [6], [7]. Additionally, disordered speech has high variability among patients. The data scarcity accelerates the insufficient generalization of ASR performance among patients [8]. Therefore, pathological speech recognition development can be rephrased by the task of developing a model with a small amount of data. In ASR, the supervised learning approach was state-of-the-art for a long time. However, supervised learning algorithms are data-hungry and require a large amount of labeled data to develop well-performing models; thus, it is not ideal for pathological speech recognition development. In order to overcome this adversity, a transfer learning approach was explored [9]–[11] where an acoustic model was first trained with a large healthy speech corpus and then fine-tuned with a dataset including pathological speech. Additionally, data augmentation techniques were exploited [12], [13] where data size was increased by adding synthetically generated pathological speech. Although both approaches showed improvements over baselines without these implementations in terms of WER, it is still far away to be competitive with ASR for healthy speech.

Recently, self-supervised learning (SSL) has been attracting many researchers

due to its remarkable success [14] in the diverse fields such as natural language processing (NLP) [15], computer vision [16], and speech processing [17]–[20]. SSL is featured as the way of learning to "obtain 'labels' from the data itself by using a 'semi-automatic' process" and "predict part of the data from other parts" [14]. In practice, SSL models are first pre-trained on unlabeled examples to figure out their universal representations, which are capable of grasping unseen data, and then fine-tuned for a specific downstream task. SSL has two advantages over traditional supervised learning: data efficiency and generalization ability [14], [21]. First, as SSL allows models to learn universal representations of data from data itself, the massive efforts of human annotations in supervised learning are no longer necessary. Second, SSL is able to learn well-generalized representations of data that can be used for multiple specific tasks (also known as downstream tasks) by exploiting the substantial amount of unlabeled data. Due to its greatness in obtaining data representations, only a few layers of the network are often added on top of SSL models for downstream tasks [15], [19]. Hence, the ability of generalization can further reduce the cost of designing the network architectures for downstream tasks. For ASR, the first powerful SSL model called wav2vec 2.0 [19] was introduced in 2020. The work was motivated to tackle the data scarcity issue in minority languages for ASR development. wav2vec 2.0 achieved the state-of-the-art performance with 1.8/3.3 WER on the widely used benchmark corpus, the Librispeech clean/other test sets [22]. Additionally, wav2vec 2.0 pre-trained on 53k hours of unlabeled data achieved 4.8/8.2 WER with only ten minutes of labeled data. The remarkable results have opened many opportunities for ASR development with limited amounts of labeled data [23], [24].

1.1 Research Question & Hypothesis

Intuitively, SSL that does not require a large amount of labeled data can be a solution to the data scarcity issue in dysarthric speech recognition. Additionally, the SSL's pre-training and fine-tuning strategy can perfectly replace the transfer learning approach seen in supervised learning. Motivated by the successful use of wav2vec 2.0 in ASR for low-resource languages [23], [24], which is a similar task to ASR for dysarthric speech where only the limited amount of data is available, this research investigates the effectiveness of SSL and wav2vec 2.0 applicability for Dutch dysarthric speech recognition. Hence, the research question and hypothesis are defined as follows.

Research Question: Can self-supervised learning outperform supervised learning for Dutch dysarthric speech recognition?

Hypothesis: Following [23], [24], it is hypothesized that self-supervised learning can outperform supervised learning for Dutch dysarthric speech recognition.

In order to answer the research question, this work first pre-trains publicly available wav2vec 2.0 with a large Dutch and Flemish corpus called Corpus Spoken Dutch (CGN) [25]. Later, the pre-trained wav2vec 2.0 and wav2vec 2.0 XLSR, which is cross-lingual wav2vec 2.0, are fine-tuned with a small amount of Flemish dysarthria speech corpus called Corpus Pathological and Normal Speech

(COPAS) [26]. Flemish is a variety of Southern Dutch that shares the same alphabet and many lexical items with standard Dutch [27]. The models are tested on the Domotica database [28], which contains voice commands utterances from Dutch dysarthria patients. The results are compared with the previously implemented DNN-HMM based supervised model [29], which follows the same training process with the same corpus. Additionally, the speaker-dependent model is also developed motivated by the results from the speaker-independent experiments. The fine-tuned wav2vec 2.0 XLSR is re-fine-tuned and tested with the target speaker's voice commands from the Domotica database.

1.2 Research Contributions

While previous works [27], [29]–[32] have addressed ASR of Dutch dysarthric speech, to the best of the author's knowledge, no research has investigated the use of SSL to those ends. Moreover, in the five aforementioned previous works, all but one [29] used datasets that were not publicly available or open access. Consequently, this hinders efforts to benchmark. Notably, the previous work that used only publicly available data [29] had a relatively different scope, the Spoken Language Understanding (SLU) task, where the model learns text-free speech-to-semantics mapping. Therefore, the detailed results of ASR experiments are not released. Additionally, the research proposes a unique SSL training strategy framework for dysarthric speech recognition. Hence, the contributions of the research can be summarized as follows:

- The work provides the first use case of SSL model for Dutch dysarthric speech recognition. This plays a crucial role as a catalyst of the SSL research in Dutch dysarthric speech recognition.
- The experiments are implemented with only publicly available data, and it is the first attempt that releases the full details of ASR results on these datasets. This allows future research to easily benchmark their results, encouraging further development of the field.
- The research outcome provides a meaningful indication of the optimal training strategy for SSL models of dysarthric speech recognition. It is summarized as the SSL training framework, which any research follows or augments.

The author believes the research can be the first step toward state-of-the-art Dutch dysarthric speech recognition with maximum feasibility.

1.3 Thesis Outline

The thesis is structured as follows. Chapter 2 provides the technical background that guides readers to understand the work better. Specifically, it focuses on the necessary knowledge to understand wav2vec 2.0. Chapter 3 introduces the related works. It covers past studies on supervised and self-supervised learning for dysarthric speech recognition. Chapter 4 describes the methodology and all details of the experiments. Chapter 5 presents the work results, and chapter 6 discusses the results to answer the research questions and the future research direction. Chapter 7 gives the conclusion with a summary of the experiments and findings from the research.

Chapter 2

Background

This chapter provides readers with sufficient knowledge to understand the work better. To those ends, the attention mechanism and Transformer model, one of the building blocks of wav2vec 2.0, is introduced first. Next, a few SSL models for audio modality, including Contrastive Predictive Coding (CPC), wav2vec, and vector quantized wav2vec (vq-wav2vec), are explained to give an overview of how SSL works. These are particularly important to understand wav2vec 2.0 since all works are the base of wav2vec 2.0. The chapter ends by explaining wav2vec 2.0 and wav2vec 2.0 XLSR, the models used for the Dutch dysarthric speech recognition.

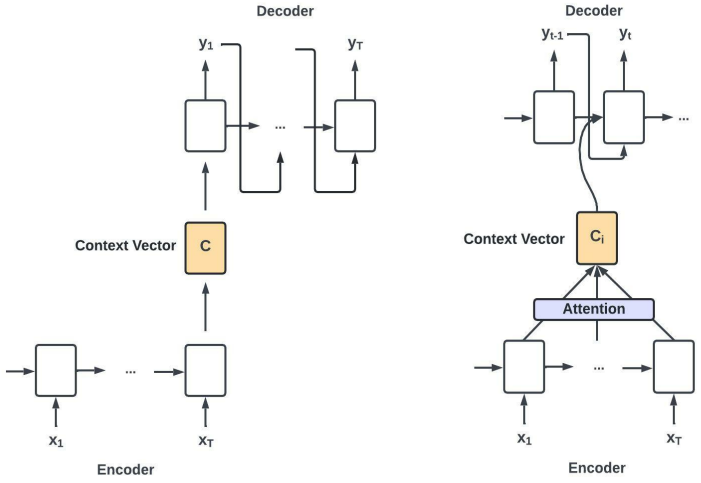
2.1 Attention Mechanism and Transformer

2.1.1 Attention Mechanism

The attention mechanism was first introduced in machine translation [33] and has been applied to many other tasks [34]–[40] due to its great success. The motivation behind the introduction of the attention mechanism was to tackle the challenge posed by the sequence-to-sequence (seq2seq) model dealing with longer sequences. By 2014, most of the architectures adopted in neural machine translation were encoder-decoder architectures [41], [42] (also know as seq2seq models) [33]. The problem with the encoder-decoder approach was the poor performance in longer sequences [43]. In order to tackle this issue, [33] added the attention module between an encoder and decoder that replaces a fixed-length context vector in conventional encoder-decoder architecture with an adaptive context vector weighted with attention weights. Fig. 2.1 illustrates the difference between traditional approach and encoder-decoder with attention mechanism.

The attention mechanism allows a decoder to select the information from the input sequence that should be attended to decode an output at the given time step [33]. The following computations were formulated to achieve this for a machine translation task. Considering an encoder-decoder architecture where the encoder comes with a bidirectional recurrent neural network (RNN) and the decoder comes with unidirectional RNN, the output y_i at time step i given the input x is computed as:

$$p(y_i|y_1, ..., y_{i-1}, X) = g(y_{i-1}, s_i, c_i). \tag{2.1}$$



(A) The traditional encoder-decoder architecture. (B) The encoder-decoder with attention mechanism.

FIGURE 2.1: Traditional encoder-decoder approach vs. encoder-decoder with attention mechanism

In equation 2.1, s_i denotes an RNN hidden state of the decoder for time step i, which is computed as:

$$s_i = f(s_{i-1}, y_{i-1}, c_i). \tag{2.2}$$

The c_i is a context vector that includes the relevant information from the encoder to decode the output at the time step i. Let hidden states of the encoder $(h_1, ..., h_{T_x})$ where each hidden state at time step j, h_j is the concatenation of bidirectional units. The context vector c_i is then computed as follows.

$$c_i = \sum_{j=1}^{T_x} \alpha_{i,j} h_j, \tag{2.3}$$

where $\alpha_{i,j}$ is an attention weight for (or the importance of) h_j with respect to the decoder's previous hidden state s_{i-1} in decoding the next hidden state s_i and output y_i. The attention weight $\alpha_{i,j}$ is computed by

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \tag{2.4}$$

which is a softmax over e_{ij} defined as

$$e_{ij} = a(s_{i-1}, hj). \tag{2.5}$$

The equation 2.5 is an alignment model/attention scoring function that computes the similarity between the inputs around the time step j and outputs around the time step i. The parameter a in the alignment model is calculated by a feedforward neural network which is jointly trained with all other modules. In practice, s_{i-1} and h_j are concatenated together into one vector to compute the alignment score; hence the attention scoring function 2.5 is known as additive attention. Fig 2.2 visualizes the computations of the decoder with attention mechanism at time step i explained above.

By incorporating the attention mechanism, the RNN-based encoder-decoder model performed well even for longer sequences as the model adaptively selects the relevant source from the input sequence to decode output at each time step [33].

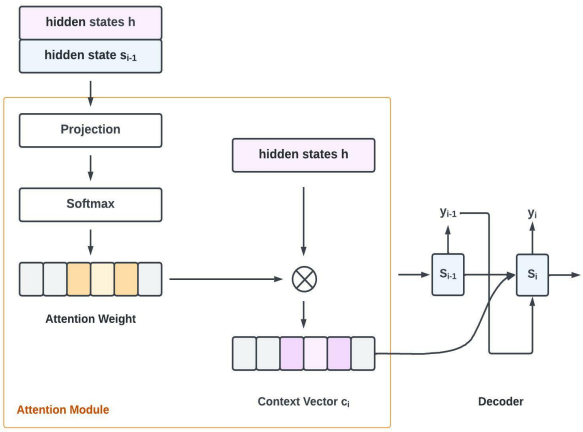


FIGURE 2.2: Decoder with attention mechanism at time step i introduced in [33].

2.1.2 Transformer

Previously, the role of the attention mechanism in sequence modeling was explained by reviewing the original work in machine translation [33] where the attention mechanism is incorporated into the RNN-based encoder-decoder architecture. This section moves on to the advanced work in attention mechanism and reviews the model named Transformer [44] that is the first model based solely on attention mechanisms. Transformer showed a significant advantage over the encoder-decoder with attention mechanism model [44] and has been applied to many other tasks [15], [19], [20], [45], [46], including wav2vec 2.0.

The motivation behind the introduction of Transformer was to tackle the sequential computation underlying issue in recurrent models [44]. Recurrent models usually compute hidden states for each time step sequentially. Due to the sequential computation, it prohibits the models from parallel computation within training examples, which casts a critical limitation on memory storage for longer sequences. Transformer was proposed to address this issue by designing the model entirely with attention mechanisms allowing parallel computation even for sequence modeling. The model architecture is delineated below by decomposing the model into several building blocks, self-attention, multi-head attention, and positional encoding, and putting them back together at the end.

Self-Attention

Self-attention is the attention mechanism adopted in Transformer. Self-attention is also called intra-attention and is explained as the attention mechanism computing undirected relations among all tokens of each input sequence [47]. Hence,

the self-attention model determines which token in the sequence should be attended to decode an output at each time step. Self-attention used in Transformer applies Scaled Dot-Product Attention [44] as an attention scoring function, which is defined by

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{2.6}$$

where T is the length of the input sequence and $\sqrt{d_k}$ is the scaling factor where d_k denotes a dimension of K. The three parameters, Q, K, and V are called as query, key, and value respectively, and all are vectors computed by linear projection described as

$$Q = W^Q X, \tag{2.7}$$
$$K = W^K X, \tag{2.8}$$
$$V = W^V X. \tag{2.9}$$

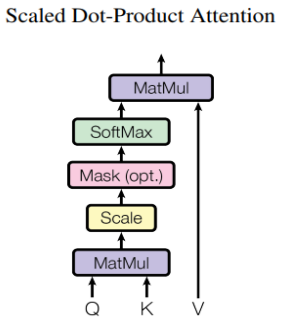


FIGURE 2.3: The overview of the vectorized self-attention computation in Transformer [44].

One way to comprehend these parameters is to consider query as a question of the event at each time step and key as a response to the question that is paired with value, which is an input sequence [48]. To describe this intuitive idea in self-attention mechanism, let the equation 2.6 be simplified by eliminating the scaling factor: then it is a dot-product attention [49]. The computation at time step i is defined as

$$Attention(q, K, V) = \sum_i^{T_x} \frac{\exp(q \cdot k_i)}{\sum_j^{T_x} \exp(q \cdot k_j)} v_i, \tag{2.10}$$

where $q \cdot k_i$ computes the attention score for k_i with respect to q, which can be considered as the appropriateness of question-answering at the time step i if borrowing the intuitive idea described above. Then, the computed attention score at each time step is normalized with the softmax function resulting in the attention weights. The attention weights are finally multiplied by v_i so that relevant information in the input sequence with respect to time step i are carried out as an output. Fig 2.4 depicts the self-attention computation at time step 2. Note that, in practice, the computation is parallel over the entire sequence. Fig 2.3 is the overview of the vectorized version of self-attention.

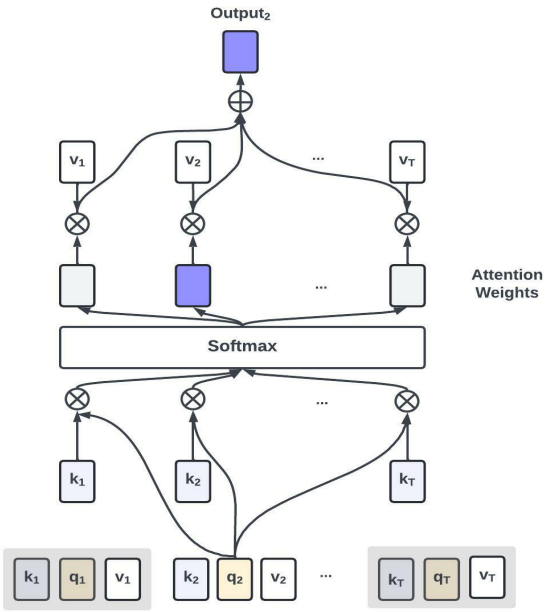


FIGURE 2.4: Self-attention computation at time step 2. The computation goes until time step T.

Multi-Head Attention

Instead of using one self-attention block, Transformer computes several heads of self-attention, called multi-head attention to obtain richer information of self-attention. The multi-head attention is computed by applying different learned linear projections for the number of heads, h, over queries, keys, and values. The multi-head attention is defined by

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., head_h)W^O \tag{2.11}$$

$$\text{where head}_i = \text{Attention}(QW_i{}^Q, KW_i{}^K, VW_i{}^V) \tag{2.12}$$

where the projections are with parameters $W_i{}^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i{}^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i{}^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ where d_{model} is dimensional keys, values and queries. Fig 2.5 is the overview of the multi-head attention.

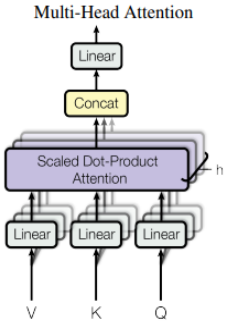


FIGURE 2.5: The overview of the multi-head attention [44].

Positional Encoding

As self-attention is the attention mechanism computing undirected relations among tokens in the sequence, the sequence order information is not considered. To keep each token's order information, [44] added positional encodings to the input embeddings. The dimension of the positional encodings is the same as the input embeddings with d_{model}. In order to compute the positional encodings, [44] used sin and cosine functions of different frequencies, which are defined as

$$PE_{(pos,2_i)} = \sin(pos/10000^{2_i/d_{\text{model}}}) \tag{2.13}$$

$$PE_{(pos,2_{i+1})} = \cos(pos/10000^{2_i/d_{\text{model}}}). \tag{2.14}$$

In the equations 2.13 and 2.14, pos denotes the position and i denotes the dimension. Hence, the positional encodings are described as a sinusoidal waveform with a geometric progression from 2π to $10000 \cdot 2\pi$.

Transformer Model Architecture

All essential components of Transformer have been explained above. Let us now consider how these components are integrated into a singular model. Transformer takes an encoder-decoder architecture. Each layer in the encoder contains two sub-layers, multi-head attention, and a position-wise fully connected feedforward network. The encoder has $N = 6$ layers in total. The multi-head attention takes keys, queries, and values from the input embeddings, which are summed with the positional encodings. The residual connections [50] are added to both sub-layers, and the normalization is applied to the output of each sub-layer. The output dimension of all sub-layers and embedding layers is fixed as $d_{\text{model}} = 512$.

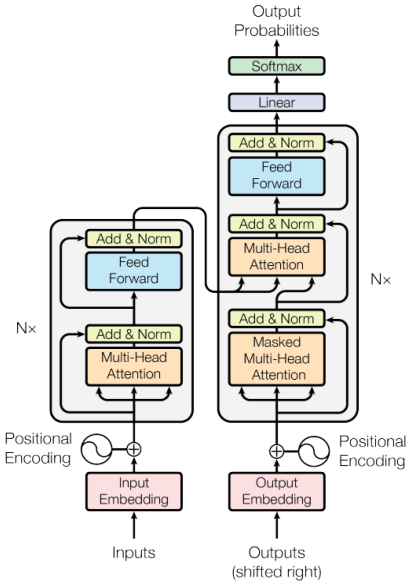


FIGURE 2.6: The Transformer model architecture [44].

The decoder has three sub-layers, two multi-head attentions, and a position-wise feedforward network per layer. The decoder takes $N = 6$ layers in total.

The residual connections and normalization are also applied similarly with the encoder. The first multi-head attention sub-layer takes queries, keys, and values from the outputs, which are also summed with the positional encodings. In order to preserve the auto-regressive inference during the decoding, the inputs at all future time steps are masked inside the scaled dot-product attention function. The second multi-head attention takes the queries from the previous layer of the decoder and the keys and values from the encoder output. Due to this information flow, the decoder is able to attend the entire input sequence. The model architecture is depicted in Fig. 2.6.

2.2 Self-supervised Learning for Audio

The previous section walked through the attention mechanisms in sequence modeling and explained the first model entirely based on the attention mechanism, named Transformer. This section proceeds to self-supervised learning (SSL). SSL allows models to learn latent feature representation from unlabeled examples. The learned model is then fine-tuned with a small amount of labeled data for a specific downstream task. The SSL's superiority is that SSL can utilize unlabeled data, which is becoming more available in the big data era. Hence, SSL can mitigate the limitation of supervised learning with the expensive manual data labeling and learn the well-generalized representation of data, resulting the less effort in designing a model architecture for a downstream task [21]. This section describes how SSL works and its superiority by reviewing several SSL models for audio modality. These reviews benefit the complete understanding of wav2vec 2.0 and wav2vec 2.0 XLSR, the models used for the experiment.

2.2.1 Contrastive Predictive Coding

Contrastive Predictive Coding (CPC) was introduced in 2019 [17]. CPC integrates the idea of predictive coding [51], [52] in signal processing to the works in NLP and computer vision, predicting a part of data from other parts [53], [54]. CPC learns the high-level latent representation, shared among different parts of input data, by predicting the future values based on the context (present) in the latent space [17]. The model was tested on different domains, including speech, images, text, and reinforcement learning in 3D environments, and achieved competitive performance in all domains.

Model Architecture

The proposed model architecture for audio representation learning consists of the encoder and autoregressive module. The encoder takes five layers strided convolutional neural network with kernel size $(10, 8, 4, 4, 4)$, strides $(5, 4, 2, 2, 2)$, and ReLU activations. The auto-regressive module is Gated Recurrent Unit (GRU) RNN. The encoder g_{enc} maps the input sequence at time step t, x_t to a sequence of latent representations $z_t = g_{\text{enc}}(x_t)$. Then, the autoregressive module g_{ar} encapsulates all past time steps latent information $z_{\leq t}$ and outputs a context latent representation $c_t = g_{\text{ar}}(z_{\leq t})$.

In order to predict time steps k in the future, the encoder g_{enc} infers z_{t+k} by encoding x_{t+k} (future time steps input) and c_t (present context) to maximize the

mutual information between the target x and context c. The computation of the mutual information is defined as

$$I(x;c) = \sum_{x,c} p(x,c) \log \frac{p(x|c)}{p(x)}. \tag{2.15}$$

In practice, the density ratio is modeled to preserve the mutual information between x_{t+k} and c_t, which is defined as

$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}, \tag{2.16}$$

where $\propto$ denotes "proportional to". For the density ratio modeling, a simple log-bilinear model is used for the prediction at each time step k, which is defined by

$$f_k(x_{t+k}, c_t) = \exp(z^T{}_{t+k} W_k c_t). \tag{2.17}$$

In linear projection $W_k^T C_t$, W_k is different for every time step k. The overall model architecture is depicted by Fig. 2.7.

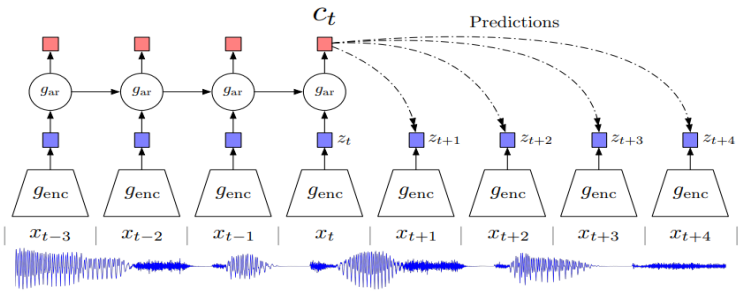


FIGURE 2.7: The overview of the CPC model architecture [17].

For a downstream task, either c_t or z_t is used. [17] suggested c_t for speech recognition task as it contains context information from the past. In the experiment, [17] used c_t for phone classification and speaker classification tasks.

Learning Objective

The learning objective is InfoNCE loss [17] inspired by Noise-Contrastive Estimation (NCE) [55], and both the encoder and auto-regressive module is jointly trained. The InfoNCE loss uses categorical cross-entropy loss to identify the positive sample x_{pos} taken from $p(x_{t+k}|c_t)$ distribution from a set of samples N including the positive and $N-1$ negative samples taken from the 'proposal' distribution $p(x_{t+k})$. The InfoNCE loss for every time step k prediction is then defined by

$$\mathcal{L}_{\text{InfoNCE}} = -\mathop{\mathbb{E}}_{X}\left[log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]. \tag{2.18}$$

Optimizing InfoNCE allows $f_k(x_{t+k}, c_t)$ to estimate the density ratio; as a result, optimize the mutual information defined by 2.15.

Connection to wav2vec 2.0

The CPC and wav2vec 2.0 share a learning objective at the abstract level. As described above, the CPC learning objective follows noise contrastive learning [14], [55] where the model tries to identify a true sample from a set of samples including a positive and $N-1$ distractors [17]. The idea of noise contrastive learning is also applied in wav2vec 2.0 with a different objective function.

2.2.2 wav2vec

wav2vec was introduced in 2019 [18] as an advanced work of CPC [17]. CPC applied the learned feature representations to phone classification, whereas wav2vec applied them to ASR. wav2vec made a few modifications to the model architecture and learning objective. However, the core idea of CPC, learning latent feature representations by predicting feature values based on the present context, is inherited by wav2vec [17]. wav2vec achieved 2.43 WER on the WSJ corpus [56], [57] nov92 test set, which was the best reported character-based system at that time.

Model Architecture

wav2vec is entirely based on convolutional architecture and consists of two networks, the encoder network, and the context network.

Encoder network: The encoder network has a five-layer causal convolution [17], which is identical to the CPC's encoder. The encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ takes a raw audio signal $x_i \in \mathcal{X}$ and projects the input to a latent feature space $\mathcal{Z}$. The output of the encoder is a latent feature representation $z_i \in \mathcal{Z}$ representing every 10ms, which covers 30ms of 16kHz of audio.

Context network: The context network consists of nine layers of causal convolution with kernel size three and stride one. The receptive field of the context network is about 210ms in total. The given receptive field v, the context network $g : \mathcal{Z} \mapsto \mathcal{C}$ summarizes the multiple latent feature representations $z_i, ..., z_{i-v}$ from the encoder network to a single contextualized tensor $c_i = g(z_i, ..., z_{i-v})$. The overview of the model architecture is described in Fig. 2.8. The output from the context network is used to train an acoustic model to do the downstream task ASR.

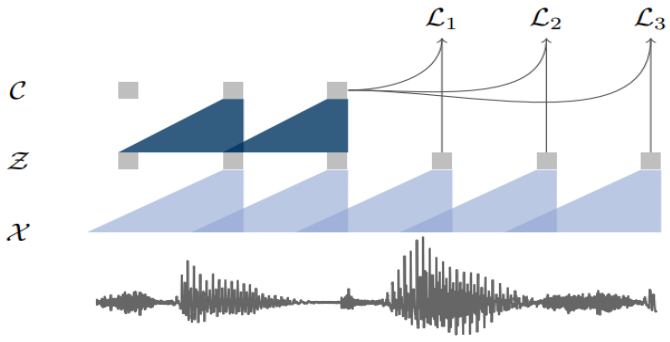


FIGURE 2.8: Overview of the wav2vec model architecture [18].

Learning Objective

The learning objective is the contrastive loss that forces the model to distinguish a true sample z_{i+k} at future k step from distractors $\tilde{z}$. The distractors are randomly obtained from each audio sequence, following a uniform distribution $p_n(z) = 1/T$, where T is the sequence length. The contrastive loss for each future step $k = 1, ..., K$ is defined by

$$\mathcal{L}_k = -\sum_{i=1}^{T-k} (\log \sigma(z_{i+k}^\top h_k(c_i)) + \lambda \mathbb{E}_{\tilde{z} \sim p_n} [\log \sigma(-\tilde{z}^\top h_k(c_i))]) \tag{2.19}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ and $\sigma(z_{i+k}^\top h_k(c_i))$ is the probability of z_{i+k} being a positive sample. λ is the number of negative samples and h_k is a linear projection seen in the equation 2.17 of CPC defined as $h_k(c_i) = W_k c_i + b_k$ where each time step k has different W_k. The model learns by minimizing the total contrastive loss $\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}_k$.

Connection to wav2vec 2.0

As shown in 2.2.1, the noise contrastive learning strategy used in wav2vec is also applied in wav2vec 2.0.

2.2.3 vq-wav2vec

vq-wav2vec was introduced in 2020 [58] as an advanced work of wav2vec. The major change from wav2vec is the discretization of speech representation in latent space, a popular approach in autoencoder [59]–[61]. The fundamental model architecture follows wav2vec. In order to perform a downstream task ASR, discretized latent speech representations from pre-trained vq-wav2vec are fed into BERT [15], which is a popular NLP self-supervised model based on Transformer. The outputs from BERT are then fed into an acoustic model to perform transcriptions prediction. The further details of BERT training will be discussed below. The proposed pipeline of ASR achieved state-of-the-art on WSJ speech recognition at that time.

Model Architecture & Learning Objective

vq-wav2vec has two convolutional networks $f : \mathcal{X} \mapsto \mathcal{Z}$ and $g : \mathcal{Z} \mapsto \hat{\mathcal{Z}}$, which are identical to the encoder network and context network introduced in wav2vec. In addition two networks, vq-wav2vec intoduces a quantization module $q : \mathcal{Z} \mapsto \hat{\mathcal{Z}}$. The overview of the model architecture is depicted by Fig. 2.9.

Encoder and context networks: The encoder network f embeds 30ms segments of the raw speech signal into a dense latent speech representation z at a 10ms stride. The encoded audio samples are fed into the quantization module q. The context network g takes the output from the quantization module to omit contextualized tensors c.

Vector quantization: The quantization module discretizes the latent speech representations z from the encoder network to $\hat{z}$. In other words, the latent speech representations z is replaced by $\hat{z} = e_i$ from a fixed size codebook $e \in \mathbb{R}^{V \times d}$. The

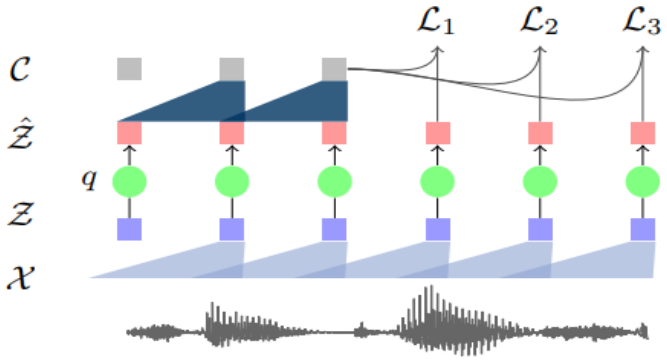


FIGURE 2.9: Overview of the vq-wav2vec model architecture [58].

codebook contains V representations with d size. In order to pick a codeword e_i from the codebook, two techniques were proposed, Gumbel-Softmax and online K-means clustering. The Gumbel-Softmax [62] allows choosing a codeword in a fully differentiable manner. The latent feature representations z are first applied linear layer with a ReLU activation followed by another linear projection. The output logits $l \in \mathbb{R}^V$ representing the codebook vectors e is then used for the Gumbel-Softmax, where the output probabilities for selecting the j-th codeword is defined as

$$p_j = \frac{\exp(l_j + n_j)/\tau}{\sum_{k=1}^{V} \exp(l_k + n_k)/\tau}. \tag{2.20}$$

In equation 2.20, τ is a non-negative temperature, $n = -\log(-\log(u))$ and u are samples from the uniform distribution $\mathcal{U}(0,1)$. During the forward propagation, the outputs with the maximum probability $i = \arg\max_j p_j$ is used while the true gradient of the Gumbel-Softmax outputs is used during the back propagation. When the Gumbel-Softmax is used for the quantization module, the learning objective remains same as the contrastive loss 2.19 (re-defined below) used in wav2vec.

$$\mathcal{L}_k^{\text{wav2vec}} = -\sum_{i=1}^{T-k} (\log \sigma(z_{i+k}^{\top} h_k(c_i)) + \lambda \mathbb{E}_{\tilde{z} \sim p_n}[\log \sigma(-\tilde{z}^{\top} h_k(c_i))]) \tag{2.21}$$

The visualization of the quantization with the Gumbel-Softmax is described by Fig. 2.10.

K-means clustering is also used as an alternative approach for the vector quantization [59]. K-means vector quantization computes the distance between input features z and the codebook vectors e and selects the closest codeword. The distance is computed by the Euclidean distance, producing $i = \arg\min_j \|z - e_j\|_2^2$, where ith codeword e_i is selected during the forward propagation. In order to obtain gradients, the learning objective is modified as follows:

$$\mathcal{L} = \sum_{k=i}^{K} \mathcal{L}_k^{wav2vec} + (\|sg(z) - \hat{z}\|^2 + \gamma \|z - sg(\hat{z})\|^2) \tag{2.22}$$

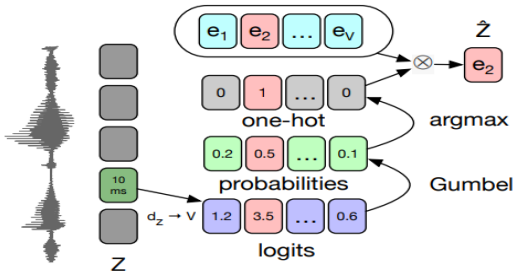


FIGURE 2.10: Visualization of the Gumbel-Softmax quantization [58].

where $sg(x) \equiv x$, $\frac{d}{dx}sg(x) \equiv 0$ is the stop gradient operator and γ is a hyperparameter. The second term $\|sg(z) - \hat{z}\|^2$ updates the codebook vectors to make them closer to the encoder output, and third term $\|z - sg(\hat{z})\|^2$ forces the encoder to produce codebook vectors representation near a centroid of the codebook. The visualization of the quantization with K-means clustering appears in Fig. 2.11.

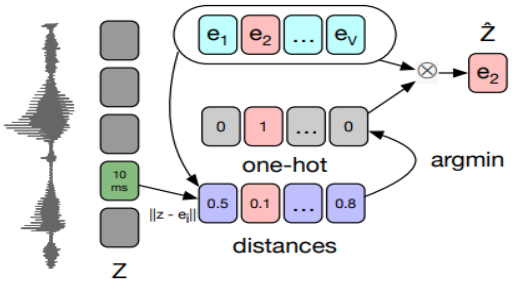


FIGURE 2.11: Visualization of the K-means quantization [58].

For the quantization module, an additional technique, multiple codebooks, was introduced to avoid mode collapse where the quantization picks only some of the codewords. The latent feature $z \in \mathbb{R}^d$ is first cast into multiple *groups* G with the matrix form $z' \in \mathbb{R}^{G \times (d/G)}$. As a result, a codebook in multiple groups can be indexed as $i \in [V]^G$, where V is the number of codewords in the indexed codebook and i_j, denoting the jth fixed codeword vector in the i_{th} codebook. For initialization, either shared codeword values across groups or independent codeword can be used. It was reported that sharing initialization resulted in more competitive performance.

Proposed Pipeline for ASR

The full pipeline for ASR with vq-wav2vec was proposed using BERT [15] and an extra acoustic model. Specifically, the quantized latent speech representations from the pre-trained vq-wav2vec are fed into BERT to obtain discretized data representations. These representations are then fed into the acoustic model to perform transcriptions prediction. Here, BERT pre-training process is briefly discussed as it is also relevant to wav2vec 2.0.

BERT is a multi-layer bidirectional Transformer encoder. The original work [15] used two unsupervised tasks to obtain the language representations. The first

is Masked LM, where input tokens are randomly masked at some percentages, and the model tries to predict the masked tokens from the surrounding context. The second is Next Sentence Prediction, where the model predicts whether the given two sentences are connected.

In the ASR pipeline, BERT is trained by the first task Masked LM. As one input token represents 10ms of audio, spanned masking over the several tokens is proposed. The input tokens, discretized latent speech representations, are masked for $M = 10$ consecutive steps. The starting token is randomly sampled from all tokens at $p = 0.05$ without replacement.

Connection to wav2vec 2.0

wav2vec 2.0 inherits three aspects. The first is the noise contrastive learning, which has been already discussed before. The second is the quantization of latent speech representations. wav2vec 2.0 also uses the Gumbel-Softmax quantization module to discretize the continuous speech representations in latent space. The last is BERT pre-training by masking. wav2vec 2.0 proposed the model itself as a full pipeline for ASR where masking training strategy is incorporated.

2.3 wav2vec 2.0 & wav2vec 2.0 XLSR

Previous sections have introduced the technical knowledge to explain wav2vec 2.0 [19] and wav2vec 2.0 XLSR [63], which are the models used to develop the Dutch dysarthric speech recognition. This section introduces them by picking up and gluing previously discussed components together. As wav2vec 2.0 XLSR is a variant of wav2vec 2.0, the main focus will be given to wav2vec 2.0. Note that here does not discuss the details of the model configurations, which are discussed in Chapter 4, but rather explains the fundamental of the model architecture and learning algorithm.

2.3.1 wav2vec 2.0

wav2vec 2.0 was introduced in 2020 as an advanced work of wav2vec [18] and vq-wav2vec [58] and the first powerful self-supervised framework for ASR. The wav2vec 2.0 training is divided into two phases, pre-training and fine-tuning. In the pre-training phase, the model learns latent speech representations in a self-supervised manner. After completing the pre-training, the model is fine-tuned with labeled data for the speech recognition task. The model achieved 1.8/3.3 WER on the LibriSpeech clean/other test setswav2vec2, which is still a competitive result against state-of-the-art [1].

Model Architecture

Wav2vec2.0 consists of three modules, feature encoder, contextualization, and quantization. Each is described presently.

Feature encoder: The feature encoder is similar to the encoder network seen in wav2vec and vq-wav2vec. It consists of several blocks of temporal convolution

with the GELU activation [64] and layer normalization [65] per layer. This module takes normalized raw waveform of speech $\mathcal{X}$ and produces latent speech representations $\mathcal{Z}$.

Contextualization: The contextualization module is similar to the context network seen in wav2vec and vq-wav2vec. It is applied after the feature encoder module. Instead of convolutional networks, the module takes N Transformer blocks with relative positional embedding, which is implemented by a convolution layer. The GELU activation and layer normalization are applied to the output of the convolution layer. This module takes the latent speech representations $\mathcal{Z}$ and produces contextualized representations $\mathcal{C}$ from the entire sequence.

Quantization: The quantization module is identical to the Gumbel-Softmax quantization seen in vq-wav2vec. It is applied after the feature encoder module. The quantization module in wav2vec 2.0 also employs multiple codebooks to avoid mode collapse. In the pre-training, given G codebooks with V entries $e \in \mathbb{R}^{V \times d/G}$, the model chooses one codeword from each codebook. The selected codewords are then concatenated, and a linear transformation is applied to obtain quantized latent speech representations q. The Gumbel-Softmax defined by the equation 2.20 can be rewritten for the multiple codebooks setting.

$$p_{g,v} = \frac{exp(l_{g,v+n_v})/\tau}{\sum_{k=1}^{V} exp(l_{g,k+n_k})/\tau} \tag{2.23}$$

where g and v denotes the coodebook number and v-th codeword for the codebook and $l \in \mathbb{R}^{G \times V}$. Same as vq-wav2vec, the outputs with the maximum probability $i = \arg\max_j p_{g,j}$ is used during the forward propagation while the true gradient of the Gumbel-Softmax outputs is used during the back propagation. The quantization module discretizes the continuous latent speech representation from the feature encoder to a finite set of values. The overview of the model architecture is described by Fig. 2.12.

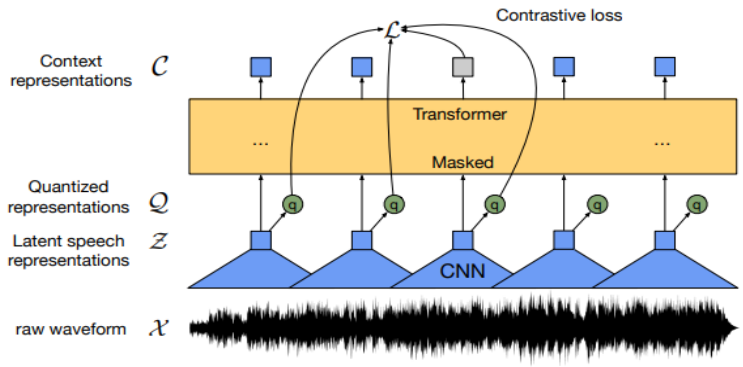


FIGURE 2.12: The overview of the wav2vec 2.0 model architecture [19].

Pre-training

One of the differences from vq-wav2vec is that wav2vec 2.0 incorporates the masking training task used in BERT into its pre-training. The masking allows the model to perform noise contrastive learning. Similar to the BERT training

seen in vq-wav2vec, the output from the feature encoder, which goes to the contextualization module, is masked for $M = 10$ consecutive time steps. During the pre-training, the model learns the speech representations by distinguishing the true quantized speech representations for a masked step from a set of distractors. The learning objective is therefore defined as:

$$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d \tag{2.24}$$

where $\mathcal{L}_m$ and $\mathcal{L}_d$ denote the contrastive loss and diversity loss respectively, and α is a tunable parameter.

The contrastive loss $\mathcal{L}_m$ is designed for the model to find the true quantized representations q_t from a set of $K+1$ quantized representations candidates $\tilde{q} \in Q_t$, for the contextualization module output c_t centered over masked time step t. The set of candidates Q_t includes the true sample q_t and K distractors, which are sampled from other masked time steps of the same audio. The contrastive loss is a negative log of the softmax over the cosine similarity between a true quantized latent feature representation and a sample from the set of candidates, which is defined as

$$\mathcal{L}_m = -log \frac{exp(sim(c_t, q_t)/k)}{\sum_{\tilde{q} \sim Q_t} exp(sim(c_t, \tilde{q})/k)} \tag{2.25}$$

where sim denotes the cosine similarity defined as $sim(a,b) = a^T b / ||a|| ||b||$. The contrastive loss in wav2vec 2.0 inherits the idea from the InfoNCE loss 2.18 in CPC. While the InfoNCE computes the similarity between a ture sample and distractors by the categorical cross-entropy, wav2vec 2.0 computes by the cosine similarity.

The diversity loss is an additional measurement against mode collapse. It is designed to nudge the model to consider all possible quantized representations in the codebooks. The diversity loss is defined as

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} = H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \; log \; \bar{p}_{g,v} \tag{2.26}$$

where $\bar{p}_g$ is each codebook in G codebooks.

Fine-tuning

In the fine-tuning phase, a linear projection is added on top of the contextualization network, and the CTC loss [66], [67] is computed for the speech recognition task. The CTC loss computes the probability distribution over all tokens that appeared in transcriptions, including language units such as phonemes, letters, and words and blank tokens indicating the character boundary. The given input audio X and output Y, the CTC loss is defined as

$$p(Y|X) \; = \; \sum_{A \in A_{X,Y}} \prod_{t=1}^{T} p_t(a_t|X) \tag{2.27}$$

where $p_t(a_t|X)$ is the probabilities at time step t. The model is trained by minimizing the negative log-likelihood instead of maximizing the likelihood directly. Hence, the final loss is defined as

$$\sum_{(X,Y)\in D} -log\ p(Y|X) \tag{2.28}$$

where D is a training set.

Additionally, a modified version of SpecAugment [68] is applied to the output from the feature encoder for the regularization purpose. The original SpecAugment applies time warping, frequency masking, and time masking to the input log-mel spectrogram. wav2vec follows the SpecAugment policy and applies the time masking same as in the pre-training and the channel masking. In order to mask channels, the starting indices are randomly selected, and 64 following channels are masked with a zero value.

Furthermore, LayerDrop [69], [70] is applied to the contextualization module. It randomly drops layers in Transformer. Note that during the fine-tuning, the quantization module is frozen, and the feature encoder is not updated.

2.3.2 wav2vec 2.0 XLSR

wav2vec 2.0 XLSR [63] is a variant of wav2vec 2.0, which has the identical model architecture and learning objective of wav2vec 2.0. wav2vec 2.0 XLSR is a model pre-trained on data from multiple languages to obtain the cross-lingual speech representations for ASR. Unlike wav2vec 2.0, the Transformer contextualization module is also updated with the extra classification layer during the fine-tuning. It was reported that the cross-lingual representations significantly outperformed the monolingual representations for ASR. The overview of wav2vec 2.0 XLSR is depicted by Fig. 2.13.

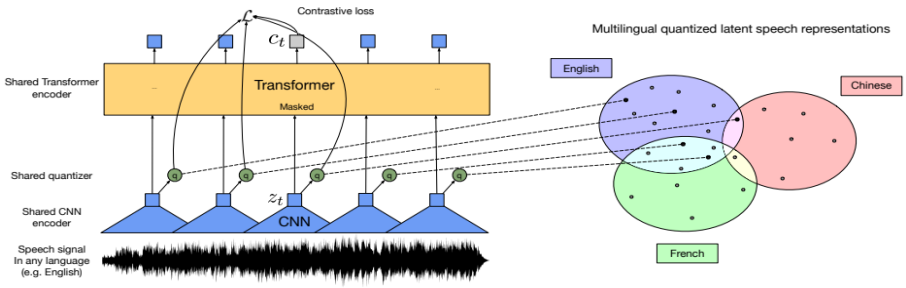


FIGURE 2.13: The overview of the wav2vec 2.0 XLSR [63].

The datasets used for wav2vec 2.0 XLSR are summarized below.

CommonVoice. The CommonVoice dataset[1] contains 38 languages with more than two thousands hours in total [71]. The following eleven languages: *English (en), Spanish (es), French (fr), Italian (it), Kyrgyz (ky), Dutch (du), Russian (ru), Swedish (sv), Turkish(tr), Tatar (tt) and Chinese (zh)*, were used for the experiment. The dataset is the November 2019 release.

[1] https://commonvoice.mozilla.org/en/languages

BABEL. The BABEL dataset[2] is a multilingual conversational telephone speech containing Asian and African languages [72]. The following ten languages: *Bengali (bn), Cantonese (zh), Georgian (ka), Haitian (ht), Kurmanji (ku), Pashto (ps), Tamil (ta), Turkish (tr), Tokpisin (tp), Vietnamese (vi)*, were used for the ptre-training. The following four languages: *Assamese (as), Tagalog (tl), Swahili (sw), Lao (lo)*, were used to evaluate cross-lingual transfer.

Multilingual LibriSpeech (MLS). The Multilingual LibriSpeech dataset [73] is a large multilingual corpus containing read audiobooks of Librivox in 8 languages: *Dutch (du), English (en), French (fr), German (de), Italian (it), Polish (pl), Portuguese (pt), Spanish (es)*. The MLS is combined with CommonVoice and BABEL to cover 53 languages and 56 thousand hours of speech. The combined dataset is used to train the model called XLSR-53.

2.4 Summary

This chapter explored the attention mechanisms to SSL models for an audio modality to fully comprehend wav2vec 2.0 and wav2vec 2.0 XLSR, which are the models used for the Dutch dysarthric speech recognition. Since SSL models are pre-trained with unlabeled data, SSL can mitigate the limitation of supervised learning with expensive manual data labeling. It also can learn the well-generalized representation of data, resulting in less effort in designing a model architecture for a downstream task. wav2vec 2.0 requires only one layer on top of the pre-trained model to achieve state-of-the-art ASR, which clearly shows the power of feature representation learning in SSL.

[2] https://catalog.ldc.upenn.edu/byyear includes es LDC2018S07, LDC2018S13, LDC2018S02, LDC2017S03, LDC2017S22, LDC2017S08, LDC2017S05, LDC2017S13, LDC2017S01, LDC2017S19, LDC2016S06, LDC2016S08, LDC2016S02, LDC2016S12, LDC2016S09, LDC2016S13, LDC2016S10.

Chapter 3

Related Works

This chapter reviews the related works of automatic speech recognition (ASR) for dysarthric speech. These reviews play a role in describing how the research questions and hypotheses are framed. The chapter is organized into three sections. Section 3.1 covers the typical approaches adopted by supervised models for dysarthric speech recognition and existing works of Dutch ASR for dysarthric speech. This section describes the research context of the supervised learning approaches and potential research exploration. Section 3.2 investigates how self-supervised learning (SSL) can be a potential approach for dysarthric speech recognition. This section gives the background of the research question and hypothesis. Finally, Section 3.3 reviews three works that have applied SSL to ASR for dysarthric speech.

3.1 Supervised Learning for Dysarthric Speech Recognition

This section reviews the supervised learning models for dysarthric speech recognition. One of the challenges for dysarthric speech recognition is a lack of available data. This creates a significant challenge for supervised learning as it requires a large amount of labeled data to be developed. Previous works [9]–[12], [27], [29]–[32] have addressed this issue by proposing two approaches: transfer learning and data augmentation. Each is described presently.

3.1.1 Transfer Learning

Transfer learning is referred to transferring the knowledge learned from the source domain to solve the new task in the target domain [48]. In dysarthric speech recognition tasks, transfer learning is often applied to compensate for the small data size. The transfer learning approach was explored [9] with speaker-dependent DNN-HMM hybrid ASR model [74] for English dysarthric ASR where an external GMM-HMM model gives the alignment. Specifically, the DNN-based acoustic model was first trained with healthy speech and then re-trained with the target data, a speaker-specific dysarthric speech. The result showed that the model with the transfer learning from healthy speech to speaker-specific dysarthric speech outperformed the model without the transfer learning.

Transfer learning has also been applied to end-to-end (E2E) models. In 2019, Google announced the project called Euphonia[1] which collected ALS patients'

[1] https://blog.google/outreach-initiatives/accessibility/impaired-speech-recognition/

dysarthric speech to develop personalized ASR. As part of the project, two experiments [10], [11] were introduced. Two pre-trained E2E models with healthy speech, called LAS [34] and RNN-T [75], were used for a personalized English dysarthric speech recognition [10]. The fine-tuned models with the target dysarthric speech achieved better performance than the model without the fine-tuning. Later, the same model, RNN-T in [10], was tested with a bigger dataset in the fine-tuning [11]. The model with only 5 minutes of fine-tuning achieved 71 % of the relative WER improvement to the model without fine-tuning and outperformed human speech recognition.

3.1.2 Data Augmentation

Data augmentation is another technique used in dysarthric speech recognition to increase the number of training samples. [13] proposed the data augmentation approach for English dysarthric speech recognition based on DNN-HMM, where dysarthric speech samples were synthetically generated from a healthy speech by temporal and speed modifications. The results showed the effectiveness of the data augmentation as the model trained on the augmented data achieved better performance. More recently, [12] investigated different data augmentation techniques, including vocal tract length perturbation, tempo perturbation, and speed perturbation for English disordered speech recognition. The results showed that the model with the data augmentation outperformed the model without the data augmentation. The speed perturbation, which modifies the audio duration and the spectral envelope, gave the best performance.

3.1.3 ASR for Dutch Dysarthric Speech

Several works have addressed ASR for Dutch dysarthric speech. All works are based on supervised learning and could fall into the transfer learning approach. The Radboud University has worked on ASR for Dutch dysarthric speech [27], [30]–[32] within the framework of the CHASING project[2]. The first work was introduced in 2016 [30] where the cross-lingual training was investigated to increase the training data size. Specifically, the DNN-HMM model was trained on the dataset containing healthy speech of Flemish (Southern Dutch) and Northern Dutch. The model was then re-trained with only Flemish healthy speech, and only the output layer was updated. The Flemish dysarthric speech data was used for testing, and the results showed that combining data among similar languages led to better dysarthric speech recognition performance. This work can also be considered a transfer learning approach, where the knowledge from the healthy speech is transferred to dysarthric speech. The second work [27] in 2017 experimented again with the transfer learning from healthy speech to dysarthric speech. The DNN-HMM model was trained on the dataset containing healthy Dutch and Flemish speech and then fine-tuned with Dutch dysarthric speech. The model outperformed the model trained only on dysarthric speech. The third experiment in 2018 [31] investigated the use of the articulatory features as an additional input. The vocal tract constriction variables obtained from speech-inversion were fed into CNN that fused with acoustic features. The output from the CNN was then fed into the DNN-HMM model. For Dutch ASR, the model was trained on either healthy or dysarthric Dutch speech and tested on dutch Dysarthric speech. For Flemish ASR, the model was trained on healthy Flemish

[2] http://hstrik.ruhosting.nl/chasing/

and Dutch speech and tested on Flemish dysarthric speech. The results showed that the use of articulatory features brought improvements in the model performance. The last experiment [32] in 2019 further investigated the use of articulatory features, gammatone filterbank (different acoustic features), and model adaptation with bottleneck features of speech with the same experiment setting of [31]. The results showed improvements with all features and the best performance using model adaptation with bottleneck features.

Recently, an ASR acoustic model was applied to Spoken Langauge Understanding (SLU) tasks for Dutch dysarthric speech [29]. As a side analysis in the experiments, the DNN-based acoustic model with pre-training and fine-tuning strategy was investigated for a Dutch dysarthric speech recognition task. The result showed that the fine-tuned model outperformed the model without fine-tuning. This research is the first attempt that utilized all public available Dutch dysarthric corpus. However, the main scope of the research was given to SLU; thus, the detailed results of the ASR task are not released. The table 3.1 summarizes the works mentioned above in Dutch dysarthric speech recognition.

	Scope	**Data**	**Approach**
[30]	ASR	Public	Flemish and Dutch speech for the training dataset & multi-stage training
[27]	ASR	Private	Further investigation of multi-stage training for the Ductch test corpus
[31]	ASR	Private	Exploration of articulatory features (vocal tract constriction variables)
[32]	ASR	Private	Exploration of gammatone filterbank & articulatory features & model adaptation with the bottleneck features
[29]	SLU	Public	Multi-stage training & model adaptation with the bottleneck features

TABLE 3.1: The list of the past Dutch dysarthric speech recognition works.

3.1.4 Insights for the Research Direction

As described above, all past works addressed Dutch dysarthric speech recognition with supervised learning where the model is trained or pre-trained on healthy speech and tested or fine-tuned with dysarthric speech. Either way, it can be considered that the model benefits from transfer learning, where the knowledge of healthy speech is transferred to dysarthric speech recognition. (Note that the transfer learning was not the only factor for the performance improvement). This approach is also adopted in English dysarthric speech recognition with E2E models by Google [10], [11]. Although all experiments showed improvement due to the transfer learning, the performance is still not competitive enough considering the benchmark ASRs for healthy speech [1]. This motivates the further exploration of dysarthric speech recognition to improve performance.

3.2 Self-supervised Learning in Low-Resource Languages

One of the research directions in dysarthric speech recognition is SSL. The task of ASR development for dysarthric speech can be rephrased by ASR development with a limited dataset. Intuitively, SSL can be applied to dysarthric speech recognition since id does not require a large amount of labeled data. Additionally, SSL's pre-training and fine-tuning learning strategy can perfectly replace the transfer learning approach adopted in supervised learning. This claim can be supported by successful SSL applications for low-resource languages, which require ASR development with a limited dataset. The past works compensated for the data limitation by the SSL pre-training with different languages. Although this causes domain mismatch between pre-training and fine-tuning, the SSL performed better than supervised learning. Pre-trained wav2vec 2.0 and wav2vec 2.0 XLSR were considered for ASR of three Indian low-resource languages, Telugu, Tamil, and Gujarati [23]. The result showed that the fine-tuned cross-lingual XLSR outperformed the supervised models of Gujarati and Tamil. Another work [24] applied pre-trained wav2vec 2.0 for various low-resource languages. The authors compared SSL and supervised pre-training with other languages for the Mandarin ASR task. The results showed that SSL pre-training could better utilize the other languages' data in terms of performance.

These successful SSL applications to low-resource languages ASR motivate the author to explore the SSL for Dutch dysarthric speech recognition. As a result, the research question is formulated as **"Can self-supervised learning outperform supervised learning for Dutch dysarthric speech recognition?"**

3.3 Self-supervised Learning for Dysarthric Speech Recognition

To the best of the author's knowledge, there are only three previous works [76]–[78] dedicated to SSL for English dysarthric speech recognition. The contrastive learning SSL was investigated for the first time with data augmentation to obtain better dysarthric speech representations [78]. The work also adopted transfer learning, where the model was pre-trained on healthy speech first. The result showed that the contrastive learning SSL outperformed the supervised pre-training. The work was further explored where different types of SSL models, wav2vec 2.0, wav2vec 2.0 XLSR, and WavLM [79] were investigated [76]. Although the potential of SSL to outperform the supervised pre-training approach was reported, the experiments were only tested on Japanese and English corpora. More recent work [77] used wav2vec 2.0, wav2vec 2.0 XLSR, and HuBERT [20] as a feature extraction module of ASR for dysarthric speech. The extracted features were fed into the acoustic model with a conformer encoder and transformer decoder to perform ASR. The performance was compared with filterbank (Fbank) features. All SSL feature extractions outperformed Fbank features, and the cross-lingual model achieved the best performance. Although the work successfully illustrated the benefit of cross-lingual pre-training, the work did not consider SSL models as a full pipeline of ASR and experimented only with English and Italian.

3.4 Summary

This chapter reviewed the past works of ASR for dysarthric speech with supervised learning and SSL. The transfer learning and data augmentation approach were typically adopted to overcome the data scarcity issue in supervised learning. The past works on Dutch dysarthric speech recognition were mainly based on transfer learning. The successful use of SSL for low-resource languages, which has a similar condition for the model development, illustrated the potential of the SSL applicability for dysarthric speech recognition. Three examples that have applied SSL for dysarthric speech recognition in different languages were introduced and described the SSL's potential benefit for the Dutch dysarthric speech recognition.

Chapter 4

Methodology

This chapter provides a methodological overview of the experiments dedicated to testing the hypothesis. The chapter is organized into two sections. Section 4.1 explains the overall strategy adopted in the research to tackle the data scarcity issue. Section 4.2 presents more detailed information on experimental settings, including dataset, experiments, and training setups.

4.1 Approach for Data Scarcity

As the previous chapter showed, this research tested the effectiveness of self-supervised learning (SSL) as an approach to the data scarcity issue in ASR for dysarthric speech. Additionally, the research also combined SSL with the data augmentation as the benefit of the combination has been reported [78].

4.1.1 Model Selection

For the experiments of SSL, the author selected wav2vec 2.0 and wav2vec 2.0 XLSR. Although WavLM [79] previously outperformed wav2vec 2.0 in dysarthric speech recognition [76], the experimental settings differ in terms of a target language and data size. Since the author's research is the first attempt to apply SSL for Dutch dysarthric speech recognition, selecting the state-of-the-art SSL model is more logical from the benchmarking perspective. wav2vec 2.0 XLSR is also considered as the benefit of cross-lingual representation for dysarthric speech recognition has been reported [77].

4.1.2 Data Augmentation

The author made use of the data augmentation technique. The combination of SSL and the data augmentation technique, named SpecAugment [68], has been already investigated [78], and the effectiveness of the data augmentation was reported. As explained in Chapter 2, SpecAugment applies time warping, frequency masking, and time masking to the input log-mel spectrogram. In the experiments, the modified version of SpecAugment [19] for wav2vec 2.0, which applies only frequency masking and time masking, was applied during the fine-tuning. The implementation is aligned with the original wav2vec 2.0 implementation.

4.2 Experimental Settings

4.2.1 Overview of the Experiments

The author first implemented Dutch dysarthric speech recognition with wav2vec 2.0 and wav2vec 2.0 XLSR to investigate how SSL works toward Dutch dysarthric speech compared to a supervised learning-based model. In order to further investigate a better SSL training strategy, the author also fine-tuned both models on the dataset, including the control speakers. The second experiment aims to analyze whether healthy speech memorization during fine-tuning improves the models' performances. Furthermore, the speaker-dependent ASR is also considered for the third experiment, motivated by the results from the previous consecutive experiments. The details of the motivation are discussed in the next chapter 5. For this experiment, the fine-tuned model on the dataset containing dysarthric and healthy speech from the second experiment is further re-fine-tuned on the target speaker. The role of the re-fine-tuning is to tailor the model to the speaker-specific speech features.

4.2.2 Dataset

The author used only publicly available Dutch dysarthria datasets for the experiments.

Pre-training Dataset

For the pre-training, the typical Dutch speech corpus, the Corpus Spoken Dutch (CGN)[1] [25] is used. The CGN contains 900 hours of Dutch speech with almost 9 million words from Flemish ad Dutch speakers. The corpus comes with the transcriptions. The author followed the past experiments for Dutch dysarthric speech recognition [27], [30]–[32] for the sub-corpus selection. Specifically, only the components of read-speech (component o), spontaneous conversations (component a), interviews (component f), and discussions (component g) are used. The resulting dataset contains 441.5 hours total, comprising 186.5 and 255 hours of Flemish and Dutch speech, respectively. Ten percent of the dataset is allocated for the validation set.

Fine-tuning Dataset

For the fine-tuning, the dysarthric Flemish speech corpus, the Corpus Pathological and Normal Speech (COPAS)[2] [26] is used. Flemish is a Dutch dialect sharing the same alphabets and a large amount of vocabulary as standard Dutch [27]. The COPAS contains speech from 122 typical Flemish speakers and 197 Flemish speakers with speech disorders, including dysarthria. The dysarthric speech sub-corpus contains 75 Flemish patients with different severity. Following past works [29], [31], [32], all annotated sentence reading tasks are extracted except the speech by patients with lower than 60 intelligibility score (IS) by Dutch Intelligibility Assessment (DIA) [80] for the fine-tuning. Specifically, the speech data is based on the 11 texts with difficulty AVI 7 or 8 (T), Text Marloes (TM,

[1]The corpus is available at https://taalmaterialen.ivdnt.org/download/tstc-corpus-gesproken-nederlands/

[2]The corpus is available at https://taalmaterialen.ivdnt.org/download/tstc-corpus-pathologische-en-normale-spraak-copas/

text with a balanced representation of Dutch phonemes), and two isolated sentences (S1 S2) are extracted from the corpus. As a data pre-processing, all audio of text tasks are separated per sentence, and the silences between sentences are removed. This reduces the duration of the audio, resulting in stable model training. The resulted dataset contains about 1-hour dysarthric speech with 214 unique sentences by 55 unique dysarthria patients. The table 4.1 below summarizes the severity statistic of the fine-tuning dataset. The Appendix A summarizes more details of the fine-tuning dataset information.

Severity (IS)	**# Speakers**	**Portion (rounded up)**
Mild (> 85)	25	45%
Moderate ($70 - 85$)	26	47%
High ($60 - 70$)	4	7%

TABLE 4.1: The severity statistics based on the intelligibility score of the fine-tuning dataset pbtained from the COPAS corpus.

The dataset is further divided into train and valid subsets. About 10 % of the whole dataset, which is about 6 minutes, is allocated for the validation. In order to ensure that the validation dataset does not contain the same sentences in the train set, all data of text numbers 6 (T6) and 11 (T11) with five different patients are used for the validation. The table 4.2 summarizes the duration of each subset.

Size	**Train**	**Valid**	**Total**
Minutes	58	6	64
Percentage	91%	9%	100%

TABLE 4.2: The dataset division for the fine-tuning.

For the second experiment, the dataset with control speakers is prepared. The healthy speech of the same test types as dysarthric speech (T, TM, S1, and S2) from the COPAS dataset is added to the dataset discussed above. The validation set is identical to the dataset without control speakers. The resulting dataset contains about 2 hours 30 minutes (1 hour of dysarthric speech and 1 hour and a half of healthy speech).

Evaluation Dataset

For the model evaluation, Domotica database[3] [28] is used. Domotica database contains voice commands speech from Dutch dysarthria patients. Typical commands in the database are "open door x" or "turn on light y". In total, it contains 38 words. The examples of the voice commands are presented in Appendix B. In order to make sure a fair comparison with the baseline model (4.2.3), the evaluation dataset contains the same data introduced in [29]. Specifically, the subcorpus Domotica 3 without two children speakers, speakers 31 and 37, is used. The speakers are categorized by severity, as the table 4.3 shows. The transcript file contains the information of each audio's condition. The audios containing

[3] https://www.esat.kuleuven.be/psi/spraak/downloads/

any distortion are excluded from the evaluation dataset. Appendix B summarizes the statistics of the evaluation dataset.

Severity (IS)	Speaker IDs
Mild (> 85)	17, 40, 43, 44, 48
Moderate ($70 - 85$)	28, 29, 34, 35, 46, 47
High ($60 - 70$)	30, 32, 33, 41

TABLE 4.3: The patients categorization in the Domotica database based on the intelligibility score [29].

Speaker-Dependent Dataset

The dataset with only a specific speaker is created for the third experiment, where the speaker-dependent ASR is developed. Dysarthric speech from a patient per severity group, ID 17, 28, and 41 in Domotica Database, are selected as target speakers for the re-fine-tuning and evaluation dataset. The patients were selected as they provide a relatively large amount of data. In order to make sure that the evaluation set does not contain the exact same voice commands that appeared in the re-fine-tuning set, the dataset is equally divided into two sets in a mutually exclusive manner, meaning that each set contains its unique voice commands. ID 40 from mild, 34 from moderate, and 30 from high severity group are also selected as the dummy target speakers for the side analysis. In the side analysis, the author tests the re-fine-tuned model on the dummy target speakers to analyze whether the model actually learns the speaker-specific features. The dataset only contains voice commands that do not appear during the target speakers' re-fine-tuning. The dataset size for each speaker is summarized in the table 4.4 and 4.5.

	ID 17	ID 28	ID 41
Minutes	25 (12)	28 (12)	18 (10)
# Voice Commands	27 (14)	27 (14)	23 (12)

TABLE 4.4: The dataset for speaker-dependent ASR experiment. The figure presented here is the total of the re-fine-tuning and evaluation sets. The figure inside of parenthesis is the number for the evaluation set.

	ID 40	ID 34	ID 30
Minutes	14	15	27
# Voice Commands	24	14	24

TABLE 4.5: The dummy target speakers evaluation dataset for the side analysis in the speaker-dependent ASR experiment.

4.2.3 Baseline Model

To judge the SSL effectiveness, the SSL models' performances are compared with the acoustic model based on a time-delay neural network (TDNN) [81] with an

external HMM-GMM model for the feature alignment. The model was developed and evaluated by [29]. The training follows the same strategy as the author's experiment with the same dataset. For pre-training, the CGN corpus was used, where Flemish data from all components except the narrow-band recordings (component c and d) and the spontaneous conversations (component a) were extracted. For the fine-tuning, all pathological speech from the COPAS dataset except the data from patients with an intelligibility score lower than 60 was used. The part of the CGN was also added to the COPAS. Hence, the more extensive dataset was used for the baseline model's fine-tuning, while the author used only the dysarthric speech of patients with an intelligibility score higher than 60 from the COPAS. The fine-tuning dataset was augmented by the speed perturbation, resulting in thrice the original training samples. The model was evaluated by the Domotica database in the manner as 4.2.2 described.

4.2.4 Tranining Setups

Below appear all the details of the training setups of wav2vec 2.0 and wav2vec 2.0 XLSR. All model training is implemented using Fairseq (Facebook AI Sequence) library[4]. The training setups are consistent among the first two experiments.

wav2vec 2.0

The feature encoder consists of seven blocks of temporal convolution taking 512 channels with strides (5,2,2,2,2,2,2) and kernel size (10,3,3,3,3,2,2). The convolutional layer for relative positional embeddings takes kernel size 128 and 16 groups.

Masking is implemented for $M = 10$ time steps from the starting point. All time steps are sampled as a starting point at $p = 0.065$ probability.

wav2vec 2.0 has two model configurations, BASE and LARGE, depending on the Transformer setup for the contextualization module. For the experiment, the BASE model is considered. The BASE consists of 12 transformer blocks, model dimension 768, inner dimension 3072 (feed-forward network), and 8 attention heads. The quantization module takes 2 groups of the codebook with the 320 codewords. The codeword size is $d/G = 128$. The Gumbel-Softmax temperature τ is varied between 2 and 0.5. The temperature in the contrastive loss 2.25 k is set to $k = 0.1$. $K = 100$ distractors are used for the contrastive loss. The diversity loss weight is set to $\alpha = 0.1$.

The pre-training configuration almost follows the original implementation. The model parameters are initialized from the pre-trained model with 960 hours of Librispeech. The maximum number of tokens per batch is set at 1400K. Adam optimizer [82] is used for the optimization. The learning rate is warmed up for 32K updates with the peak at 5×10^{-4} and then decayed with polynomial decay. The model is run until reaching 200K updates. The model with the last checkpoint is used for fine-tuning.

[4] https://github.com/facebookresearch/fairseq

For the fine-tuning, the randomly initialized output layer is added on top of the Transformer. The model is trained with the CTC loss [66]. The fine-tuning configuration also follows the original implementation. Adam and tri-state learning rate scheduler are used for the optimization. The first 10% of updates are warmed up, and the next 40% are kept constant. The rest is linearly decayed. The peak learning rate is set to 5×10^{-5}. For the data augmentation, the time masking probability is set to 0.65 with 10 steps length. The channel masking probability is set to 0.25 with 64 channel length. LayerDrop is also applied with a rate of 0.05. The model is fine-tuned up to 27K updates. Appendix C visualizes the loss and accuracy movement during the pre-training and fine-tuning of the experiments with and without control speakers.

wav2vec 2.0 XLSR

There are different types of cross-lingual wav2vec 2.0 depending on the number of languages included during the pre-training. This experiment considered XLSR-53, the LARGE model pre-trained with 56 thousand hours of speech in 53 languages. The LARGE model consists of 24 Transformer blocks with 1024 model dimension, 4096 inner dimension, and 16 attention heads. For the quantization module, the codebook size is set to $d/G = 384$.

For the fine-tuning, the randomly initialized output layer on top of the Transformer of the pre-trained XLSR-53 is added. The fine-tuning configuration almost follows the original implementation. The model is optimized by Adam optimizer with the tri-state learning rate scheduler. The first 10% of updates are warmed up, and the next 40% are kept constant. The rest is linearly decayed. The peak learning rate is set to 2×10^{-5}. The data augmentation and LayerDrop parameters are set as identical to the wav2vec 2.0 BASE model fine-tuning. The model is fine-tuned up to 16K updates. Appendix D visualizes the loss and accuracy movement during the pre-training and fine-tuning of the first and second experiments.

Speaker-Dependent ASR

In the experiment of speaker-dependent ASR, the fine-tuned XLSR-53 from the second experiment is loaded to re-fine-tune the model on the specific speaker utterances. The model with the last checkpoint is used for re-fine-tuning. From the previous fine-tuning setups, only the learning rate peak setup is modified from 2×10^{-5} to 1×10^{-5}. The model is updated for further 8000 steps. Appendix E visualizes the loss and the accuracy movement during the re-fine-tuning.

Inference

Both models are evaluated with the CTC decoder with the beam search with beam width 50. No language model is provided. The pyctcdecoder[5] library is used for the implementation.

[5] https://github.com/kensho-technologies/pyctcdecode

The fine-tuned models are available at the GitHub repository[6] with the evaluation dataset. The evaluation can be reproduced following the repository's instructions.

4.3 Summary

The chapter describes the research methodology, from the general approach to the data scarcity issue to the details of the experiments. The author used only publicly available Dutch dysarthric speech corpus for the experiments. The research implemented three experiments. First, the author developed wav2vec 2.0 and wav2vec 2.0 XLSR for Dutch dysarthric speech recognition. Following the first experiment, the author also fine-tuned these models with the dataset, including control speakers, to analyze the effectiveness of healthy speech memorization during the fine-tuning phase. Motivated by the results from the first and second experiments, the speaker-dependent ASR is also considered. The fine-tuned model from the second experiment is re-fine-tuned to make the model speaker-dependent.

[6] https://github.com/Tatsu1020/self-supervised-dutch-dysarthria-asr

Chapter 5

Results

This chapter presents the results of the experiments. In addition to the observations from the results, the indications from the observations are also discussed.

5.1 Self-Supervised Learning vs. Supervised Learning

5.1.1 Observations

First, the author looks at the results from the first experiment. In the experiment, the pre-trained wav2vec 2.0 BASE and wav2vec 2.0 XLSR-53 are fine-tuned on Flemish (Southern Dutch) dysarthric speech from the COPAS dataset. Tables from 5.1 to 5.3 present the fine-tuning results and compare the performance between the supervised baseline model and proposed SSL models. Note that the author requested access to the baseline model's WERs, which are now publicly available at the GitHub repository[1] provided by [29].

Models	ID 17	ID 40	ID 43	ID 44	ID 48	Avg.
Baseline	**37.9**	-	**30.74**	-	**26.14**	**31.59**
wav2vec 2.0 BASE	79.86	76.45	74.25	73.72	77.73	76.40
XLSR-53	40.09	**46.90**	34.93	**23.23**	34.36	35.90

TABLE 5.1: The WER per patient in the mild severity group (IS: 85 <). The model is fine-tuned without control speakers.

Models	ID 28	ID 29	ID 34	ID 35	ID 46	ID 47	Avg.
Baseline	52.80	-	**37.34**	-	**28.71**	-	**39.61**
wav2vec 2.0 BASE	85.68	73.43	81.88	75.52	77.42	76.08	78.33
XLSR-53	**52.57**	**38.91**	49.39	**35.60**	34.47	**39.92**	41.81

TABLE 5.2: The WER per patient in the moderate severity group (IS: 70 - 85). The models are fine-tuned without control speakers.

First of all, both SSL models could not outperform the baseline in most cases. Although XLSR-53 outperformed the baseline for the high severity group, the average WER remained high. For the SSL models, the cross-lingual XLSR-53 outperformed mono-lingual (Dutch in this case) wav2vec 2.0 BASE for all patients.

[1] https://github.com/wangpuup/Pre-training-with-dysarthric-speech

Models	ID 30	ID 32	ID 33	ID 41	Avg.
Baseline	-	-	71.14	65.95	68.54
wav2vec 2.0 BASE	95.51	77.71	98.13	75.20	86.63
XLSR-53	**53.76**	**38.79**	**59.57**	**49.82**	**50.48**

TABLE 5.3: The WER per patient in the high severity group (IS: 60 - 70). The models are fine-tuned without control speakers.

For the comparison among the different severity groups, the performance became worse as the severity went higher for all models. However, it is worth noting that XLSR-53 did not significantly increase WER from the mild to high severity group compared to the baseline model. XLSR-53 showed an approximate 40.61% increase in WER from the mild to high severity group, while the baseline increased about 116%.

The author also considers the performance variability among the patients within each severity group. As table 5.4 summarizes, the XLSR-53 showed a lower range than the baseline model except for the high severity group. The author intentionally eliminated the wav2vec 2.0 BASE from the comparison since the model's performance was not competitive enough to discuss the WER variability among patients (e.g., the model learned nothing might perform similarly on different speech at a poor accuracy level.) For a fair comparison, the WER for the patients not used for the baseline model evaluation is excluded from the range calculation.

Models	Mild	Moderate	High	Avg.
Baseline	11.76	24.09	**5.19**	37.58
XLSR-53	**5.73**	**18.1**	9.75	**11.19**

TABLE 5.4: The **WER range** (in points) among the patients within each severity group.

To summarize the observations, all models, which are pre-trained on Dutch healthy speech, could not achieve the competitive WER even for the mild severity group. XLSR-53 outperformed wav2vec 2.0 BASE for all patients. Although XLSR-53 showed a relatively competitive performance with the baseline model, the WERs were higher for most patients. XLSR-53 had a lower range in WER among and within the severity groups.

5.1.2 Indications from the Observations

From the observations discussed above, the following can be claimed.

1. **Cross-lingual representations benefit from the domain adaptation.** XLSR-53 outperformed wav2vec 2.0 BASE for all patients with different levels of severity. This indicates that cross-lingual representation learning in the pre-training benefits the domain adaptation from typical to dysarthric speech in the fine-tuning. The cross-lingual representations superiority

can be explained by the model's wide range of knowledge of phonetic features. Learning the different types of phonetic features during the pre-training might allow better modeling of the undiscovered domain data, dysarthric speech, during the fine-tuning.

2. **SSL model has a better generalization ability.**
XLSR-53 showed lower performance variability among and within the severity groups. This indicates that XLSR-53 has a better feature generalization ability. This is especially clear by the result that the XLSR-53 had less performance reduction from the mild to high severity group with about 25 points differences from the baseline model. The higher ability of feature generalization potentially degrades the performance when a wide variety of dysarthric speech features are learned at the same time. Hence, the further indication is that the speaker-dependent model might be an approach to developing a working-level ASR.

5.2 Effectiveness of Control Speakers

5.2.1 Observations

Next, the author examined the results from the second experiment, where the effectiveness of healthy speech memorization during fine-tuning is analyzed. The pre-trained models were fine-tuned in the same settings as the first experiment; however, the fine-tuning dataset contained healthy speech in this experiment. Tables from 5.5 to 5.7 present the WER per patient in each severity group.

Models	Fine-tune	ID 17	ID 40	ID 43	ID 44	ID 48	Avg.
wav2vec 2.0 BASE	D	79.86	76.45	74.25	73.72	77.73	76.40
wav2vec 2.0 BASE	D + H	**70.31**	**69.36**	**68.58**	**68.13**	**72.05**	**69.68**
XLSR-53	D	40.09	46.90	34.93	23.23	34.36	35.90
XLSR-53	D + H	**35.19**	**41.18**	**29.22**	**18.15**	**33.35**	**31.41**

TABLE 5.5: The WER per patient in the mild severity group (IS: > 85). The "Fine-tune" column indicates the used fine-tuning dataset. "D" denotes dysathric speech and "H" denotes healthy speech.

Models	Fine-tune	ID 28	ID 29	ID 34	ID 35	ID 46	ID 47	Avg.
wav2vec 2.0 BASE	D	85.68	73.43	81.88	75.52	77.42	76.08	78.33
wav2vec 2.0 BASE	D + H	**72.80**	**69.24**	**72.56**	**67.37**	**72.58**	**68.32**	**70.60**
XLSR-53	D	52.57	38.91	49.39	35.60	34.47	39.92	41.81
XLSR-53	D + H	**49.87**	**32.87**	**46.70**	**29.66**	**28.83**	**36.74**	**39.95**

TABLE 5.6: The WER per patient in the moderate severity group (IS: 70 - 85). The "Fine-tune" column indicates the used fine-tuning dataset. "D" denotes dysathric speech and "H" denotes healthy speech.

As tables show, both SSL models improved their performance when the fine-tuning dataset included healthy speech for all patients except the XLSR-53 WER on patient 30. XLSR-53 again outperformed wav2vec 2.0 BASE for all patients.

Models	Fine-tune	ID 30	ID 32	ID 33	ID 41	Avg.
wav2vec 2.0 BASE	D	95.51	77.71	98.13	75.20	86.63
wav2vec 2.0 BASE	D + H	**76.10**	**66.06**	**80.20**	**71.59**	**74.01**
XLSR-53	D	**53.76**	38.79	59.57	49.82	50.48
XLSR-53	D + H	54.31	**33.44**	**58.32**	**47.91**	**50.13**

TABLE 5.7: The WER per patient in the high severity group (IS: 60 - 70). The "Fine-tune" column indicates the used fine-tuning dataset. "D" denotes dysathric speech and "H" denotes healthy speech.

Additionally, XLSR-53 performance improved to a more competitive level where the WERs on some patients were lower than the baseline model.

The author also examined how the improvement of XLSR-53 differs among the severity groups. wav2vec 2.0 BASE is excluded in this analysis since it can give a bias from the highly-poor performance achieved in the first experiment. As table 5.8 shows, the performance improvement is more significant in the lower severity group. This is logical considering dysarthric speech with lower severity is more similar to healthy speech. The healthy speech memorization barely improved the performance of highly-severed dysarthric speech.

Model	Mild	Moderate	High
XLSR-53	12.50%	4.44%	0.69%

TABLE 5.8: The **WER relative improvements** by the fine-tuning with control speakers at average per severity group.

5.2.2 Indications from the Observations

Based on the observations, it is concluded that memorizing healthy speech during the fine-tuning is beneficial, although the improvement on the highly severed dysarthric speech is limited to small. As Chapter 2 explains, the SSL model learns high-level underlying feature representations during the pre-training, while it learns lower-level data-specific features during the fine-tuning. Hence, the author considers that adding control speakers to the fine-tuning dataset could encourage learning more fined granularity of healthy speech features, allowing the model to predict disorder-invariant transcripts better.

5.3 Speaker-Dependent ASR for Dysarthric Speech

The author further examined the SSL applicability to dysarthric speech recognition by experimenting with speaker-dependent ASR. The first experiment observed that the SSL model has a better generalization ability among different levels of severity. However, this superiority might not be ideal regarding dysarthric speech recognition since dysarthric speech features significantly vary from speaker to speaker. The patient with higher severity will have less intelligible speech than the patient with lower severity. Thus, fine-tuning with a speaker-severity-mixed dataset potentially degraded the model's performance on the disordered

speech of all levels of severity. From this assumption drawn from the first experiment, the author considered the speaker-dependent ASR as an alternative approach for dysarthric speech recognition. Although the severity-dependent ASR is another potential approach, the speaker-dependent ASR was considered due to the observed variability within each severity group.

5.3.1 Observations

In the experiment, the fine-tuned XLSR-53 model with a speaker-severity-mixed dataset was re-fine-tuned on the target speaker's speech to tailor the model to that speaker. The author selected the model from the second experiment since it showed better performance due to healthy speech memorization. As a target speaker, one patient from each severity group was chosen. Table 5.9 compares the speaker-independent and dependent XLSR-53 performance on different patients' speech.

Model	**Re-fine-tune**	**Mild** ID 17	**Moderate** ID 28	**High** ID 41
XLSR-53	No	35.19	49.87	47.91
XLSR-53	Yes	**10.79**	**15.17**	**17.36**

TABLE 5.9: The WER comparison between the speaker-independent and speaker-dependent XLSR-53 models. The both models are fine-tuned on the speaker-severity-mixed dataset with control speakers. "Re-fine-tune" indicates whether the model is re-fine-tuned on the target speaker.

As the results show, the re-fine-tuning of the target speakers significantly improved the model at all severity levels. It should be noted that even for the highly disordered speech, the model achieved WER lower than 20. As 5.10 presents, the improvement is more significant for the mild and moderate compared to the high severity group. **The author emphasizes that the remarkable improvement is achieved by only about 10 minutes of re-fine-tuning. This is highly encouraging for the ASR development for dysarthric speech since it provides a way to remove a strict constraint imposed by the data limitation.**

Model	**Mild** ID 17	**Moderate** ID 28	**High** ID 41	**Avg.**
XLSR-53	69.33 %	69.58 %	63.76 %	67.55 %

TABLE 5.10: The **WER relative improvement** by the speaker-dependent re-fine-tuning for each severity group.

However, one needs to be aware that the improvement might come from the domain shift in the re-fine-tuning from text read-speech to voice commands utterances. While the model sees voice commands-like utterances for the first time in the evaluation in the second experiment, the -re-fine-tuned model obtains the voice-commands speech knowledge before the evaluation. Hence, the re-fine-tuned models are tested on the dummy target speakers to analyze whether the improvement comes from the speaker-dependent training. Tables 5.11 to 5.14

summarizes the analysis.

Model	Re-fine-tune	Test ID: 17	Test ID: 40
XLSR-53	No	35.19	41.18
XLSR-53	Yes	**10.79**	35.19

TABLE 5.11: The WER improvement analysis by testing on the target speaker and dummy target speaker. "Re-fine-tune" indicates whether the model is re-fine-tuned on the target speaker, ID 17.

Model	Re-fine-tune	Test ID: 28	Test ID: 34
XLSR-53	No	49.87	46.70
XLSR-53	Yes	**15.17**	22.25

TABLE 5.12: The WER improvement analysis by testing on the target speaker and dummy target speaker. "Re-fine-tune" indicates whether the model is re-fine-tuned on the target speaker, ID 28.

Model	Re-fine-tune	Test ID: 41	Test ID: 30
XLSR-53	No	47.91	54.31
XLSR-53	Yes	**17.36**	33.00

TABLE 5.13: The WER improvement analysis by testing on the target speaker and dummy target speaker. "Re-fine-tune" indicates whether the model is re-fine-tuned on the target speaker, ID 41.

Model	Mild ID 40	Moderate ID 34	High ID 30	Avg.
XLSR-53	14.54 %	52.35 %	39.23 %	35.37 %

TABLE 5.14: TThe **WER relative improvement** by the cross-speaker adaptation with the re-fine-tuning for each severity group.

Although the model improved even for the dummy target speakers, the improvement is limited to small, and the resulted WERs are all higher than 20, which are not acceptable. The average improvement for dummy target speakers is 35.37 %, while the average improvement for the target speakers is 67.55 %, which clearly describes the benefit of speaker-specific features learning.

5.3.2 Indications from the Observations

The experiment observed that the speaker-dependent ASR with a re-fine-tuning strategy significantly improved over the speaker-independent ASR. Additionally, from the results of the side analysis, it can be concluded that the further re-fine-tuning with the target speakers indeed allows the model to learn speaker-specific features, boosting the ASR performance. The experiment demonstrated

that this could be achieved by only about 10 minutes of utterances from the target speaker. The results motivate the author to propose the new SSL training strategy, which is discussed in the next chapter 6.

5.4 Summary

The chapter presents the results of the experiments. The results showed that the SSL models could not outperform the supervised baseline model for most patients. The cross-lingual representation pre-training gave a better performance than the mono-lingual representation pre-training. The performance improved for all patients when the models were fine-tuned on the dataset with control speakers. XLSR-53 achieved relatively competitive performance with the baseline model. Additionally, XLSR-53 showed a better generalization in its performance among the different severity groups and patients. Since it potentially degraded the model's performance, the speaker-dependent ASR was also developed. The results showed that the re-fine-tuning of the target speaker with only about 10 minutes of utterances significantly improved the model's performance. The results are highly encouraging for dysarthric speech recognition development as it potentially eliminates the strict constraint imposed by the data scarcity.

Chapter 6

Discussion

This chapter revisits the research question and hypothesis and outlines future research. The research question was defined as **"can self-supervised learning outperform supervised learning for Dutch dysarthric speech recognition?"** The hypothesis was defined as **"following [23], [24], it is hypothesized that self-supervised learning can outperform supervised learning for Dutch dysarthric speech recognition."**

6.1 Did Self-Supervised Learning Outperform Supervised Learning for Dysarthric Speech Recognition?

The research investigated the effectiveness and applicability of SSL to ASR for Dutch dysarthric speech. As the previous chapter presented, the SSL models, wav2vec 2.0 BASE and wav2vec 2.0 XLSR-53 could not outperform the supervised DNN-HMM baseline model in most cases. The performance of wav2vec 2.0 was particularly far away from the competitive level of the baseline model. XLSR-53 outperformed the baseline for all patients in the high severity group. When the models were fine-tuned on the dataset, including control speakers, XLSR-53 achieved a competitive performance with the baseline model.

From the obtained results, it can not be concluded that the SSL benefits Dutch dysarthric speech recognition more than supervised learning. However, it is imperative to note that **this conclusion can only apply to the case when the SSL and supervised learning follow the training strategy, where the models are fine-tuned with speaker-severity-mixed dataset**.

The author obtained critical indication from the results in the first experiment. It is observed that SSL has an advantage in generalization ability. The XLSR-53 model showed a lower range in WER among and within the different severity groups. Hence, this potentially degraded the performance of the SSL models as dysarthric speech features greatly differ from speaker to speaker. This means that the SSL might require an SSL-specific better training strategy where the fine-tuning dataset is formulated with less variability, such as per severity group or speaker dataset.

Motivated by the indication, the author also implemented the speaker-dependent ASR. The results showed that speaker-specific feature learning by the re-fine-tuning on the target speaker significantly improved the model performance. The side analysis also confirmed that the improvement mainly came from speaker-specific features learning. Hence, the results validate the assumption drawn

from the severity-speaker-mixed fine-tuning experiment, claiming that the SSL's poor performance is ascribed to its superior generalization ability. This implies that the severity-speaker-mixed fine-tuning is not an optimal training approach for SSL, and the SSL might outperform supervised learning in this training approach.

The speaker-dependent ASR results also could explain why the hypothesis is rejected. The hypothesis was based on the results from the successful SSL applications in low-resource languages ASR [23], [24], which shares the same challenge, the data scarcity. The difference between low-resource languages and dysarthric speech is the nature of the data. While low-resource language feature variability is limited to the speaker varieties, dysarthric speech features differ depending on the type of disorder, severity level, and speaker characteristics, bringing a more diverse feature distribution. Due to SSL's generalization excellence, more attention must be paid to the dataset distribution for dysarthric speech recognition. Hence, it is valid to consider that transferring the same SSL training approach in low-resource languages to dysarthric speech recognition might not yield the same level of performance.

6.2 Effective Training Strategy of Self-Supervised Learning for Dysarthric Speech Recognition

The author extensively analyzed the SSL applicability to Dutch dysarthric speech recognition through three consecutive experiments. The first experiment demonstrated the effectiveness of cross-lingual representation learning during the pre-training. The second experiment showed the benefit of healthy speech memorization during fine-tuning. The third experiment demonstrated the effectiveness of the re-fine-tuning with the target speaker's speech for dysarthric speech recognition. It is also highly feasible since only about 10 minutes of utterances from the target speaker are required to boost the model's performance. By combining all findings from the research, the author proposes the following training strategy framework with an intuitive explanation for future research on SSL for dysarthric speech recognition.

1. **Pre-training on cross-lingual healthy speech**
   In this phase, SSL models learn a wide range of the high-level acoustic features underlying different languages.

2. **Fine-tuning on speaker-severity-mixed dysarthric speech with control speakers**
   In this phase, SSL models adapt their knowledge from healthy speech to dysarthric speech while memorizing the low-level healthy speech features. This would help to predict the disorder-invariant transcription better.

3. **Fine-tuning on the target dysarthria speaker's speech**
   Finally, the model knowledge is re-adapted to the more detailed speaker-specific speech features. Since it is assumed that the model already learns the low-level healthy speech features in the previous phase, the re-fine-tuned is implemented with only the target speaker's speech.

6.3 Future Research

- **Different SSL Models Exploration**
  The exploration of the different model architectures could be one direction. This research selected the state-of-the-art SSL model, wav2vec 2.0, for a benchmarking purpose. However, other model architectures might work better for dysarthric speech recognition. As Chapter 3.3 introduced, the better performance of WavLM [79] than wav2vec 2.0 for English dysarthric speech has been reported [76].

- **SSL Model as a Feature Extraction Module**
  The research added only one output layer on top of the transformer contextualization module for the ASR task. However, adding a more complicated acoustic model is also possible, potentially allowing more complex mapping from speech to text. As discussed before, the SSL models has been investigated as a feature extraction module [76], [77].

- **Cross-Lingual Fine-tuning**
  Previously, the author proposed the SSL speaker-dependent ASR training framework. Phase 2 in the framework could be augmented by a cross-lingual dysarthric dataset. The initial fine-tuning allows the model to adapt its knowledge from typical to dysarthric speech. The fine-tuning with cross-lingual data might improve the performance due to the knowledge acquisition of the broader range of dysarthric speech features. This is motivated by observed cross-lingual pre-training effectiveness.

- **Severity-Dependent Model**
  Within the proposed training framework, the severity-dependent model can also be an option in phase 3. Instead of re-fine-tuning on the target speaker, the model might be re-fine-tuned on different speakers from the specific severity group. Although it might degrade the performance due to patient variability, the model can cover more patients than the speaker-dependent ASR. The generalization and performance trade-off will be the research scope of this topic.

Chapter 7

Conclusion

The research explored self-supervised learning (SSL) for Dutch dysarthric speech recognition as an approach to the data scarcity issue. SSL has been successfully applied to the ASR for low-resource languages, which also requires developing models with limited data. Although three previous works have applied SSL to dysarthric speech recognition in other languages, there is no research addressing with Dutch corpus. For benchmarking, the experiment used the state-of-the-art SSL model, wav2vec 2.0.

The results showed that the healthy speech pre-training and dysarthric speech fine-tuning for wav2vec 2.0 and XLSR-53 could not yield better performance than for the supervised DNN-HMM model. For the comparison among the SSL models, the cross-lingual pre-training produced a lower WER than the mono-lingual pre-training. Additionally, the healthy speech features memorization during the fine-tuning by adding the control speakers ameliorated the models' performance.

Although the SSL models could not outperform the supervised model, the SSL's more remarkable generalization ability can explain the poor performance. The performance degradation of the SSL models from the mild to high severity group was not as significant as in the supervised model. Thus, the research took a further step by developing the speaker-dependent ASR model to analyze whether less variability in the dataset could improve the performance. The results showed the SSL's outstanding speaker-adaptation ability. The re-fine-tuning with only about 10 minutes of the target speaker significantly improved the model's performance, where the WERs for all tested patients are lower than 20.

Based on the outcome of the research, the author proposed the SSL training framework for dysarthric speech recognition. The framework suggests that an SSL model first pre-trained on cross-lingual healthy speech and fine-tuned on the dysarthric speech dataset with control speakers. As the initial fine-tuning can be considered typical-to-dysarthric speech features adaptation, the author also mentioned the cross-lingual dysarthric speech dataset as an alternative approach for future research. The model is finally re-fine-tuned for the target speaker with a small amount of data. The severity-dependent model can also be considered at this phase. However, the generalization and performance trade-off should be analyzed as the experimental result demonstrated the performance variability within the severity group.

The research is the first attempt to apply the SSL for Dutch dysarthric speech recognition. All the experiments are completed using only publicly available

data for the benchmark convenience for future research. Additionally, the research outcome proved that the SSL could be used in a speaker-dependent manner, where no significant amount of dysarthric speech data by the target speaker is required. This can remove the strict data limitation constraint in the ASR development for dysarthric speech since the speaker-mixed dataset is available more than the speaker-specific dataset. The author believes this research is the imperative milestone to developing a working-level Dutch dysarthric speech recognition and hopes the research outcome catalyzes future research.

Appendix A

Fine-tuning Dataset Statistics

Sub-corpus Name		# Speakers	# Sentences
TM (Text Marloes)		46	8
T (Text)	T1	5	20
	T2	2	12
	T3	2	14
	T4	0	34
	T5	2	15
	T6	4	17
	T7	2	19
	T8	3	16
	T9	1	19
	T10	2	14
	T11	1	24
S1 (Sentence 1)		46	1
S2 (Sentence 2)		46	1
Total		55	214

TABLE A.1: The dysarthric speech fine-tuning dataset from the COPAS.

Appendix B

Evaluation Dataset

Patient ID	Intelligibility Score	# Commands	# Utterances	Duration (mins)
17	88.6	27	347	25
28	73.1	27	204	28
29	73.6	25	174	17
30	69	27	198	37
32	65.6	22	41	19
33	66.2	10	113	27
34	76.2	27	331	32
35	72.3	27	268	30
40	85.5	27	184	18
41	64.2	23	144	26
43	89.4	10	133	11
44	89.2	28	164	15
46	74.9	10	97	12
47	73.4	24	64	10
48	85.8	10	169	17

TABLE B.1: The statistics of the evaluation dataset from the Domotica database [28].

Voice Command
ALADIN LICHTEN IN DE WOONKAMER EN KEUKEN UIT
ALADIN AL DE LICHTEN UIT
ALADIN DEUR DICHT VAN BADKAMER
ALADIN DEUR BADKAMER OPEN
ALADIN LICHT IN DE BADKAMER AAN
ALADIN HOOFDEINDE OP STAND EEN
ALADIN HOOFDEINDE OP STAND TWEE
ALADIN HOOFDEINDE OP STAND DRIE
ALADIN LICHT IN DE KEUKEN AAN
ALADIN LEESLAMPJE AAN
ALADIN ROLLUIKEN SLAAPKAMER NEER
ALADIN ROLLUIKEN SLAAPKAMER OMHOOG
ALADIN ROLLUIK ACHTERDEUR NAAR BENEDEN
ALADIN ROLLUIK OMHOOG DEUR IN LIVING ALADIN ROLLUIKEN NEER
ALADIN ROLLUIKEN OMHOOG IN LIVING
ALADIN LICHT AAN SLAAPKAMER
ALADIN LICHT UIT IN SLAAPKAMER
ALADIN STAANDE LAMP OP EEN
ALADIN STAANDE LAMP OP TWEE
ALADIN STAANDE LAMP OP DRIE
ALADIN THERMOSTAAT CHAUFFAGE OP EENENTWINTIG
ALADIN DE VOORDEUR TOE
ALADIN DEUR OPEN
ALADIN DEUR SLAAPKAMER DICHT
ALADIN DEUR SLAAPKAMER OPEN
ALADIN IN DE WOONKAMER LICHT AAN

TABLE B.2: The voice command examples from the evaluation dataset.

Appendix C

wav2vec 2.0 Loss Movement

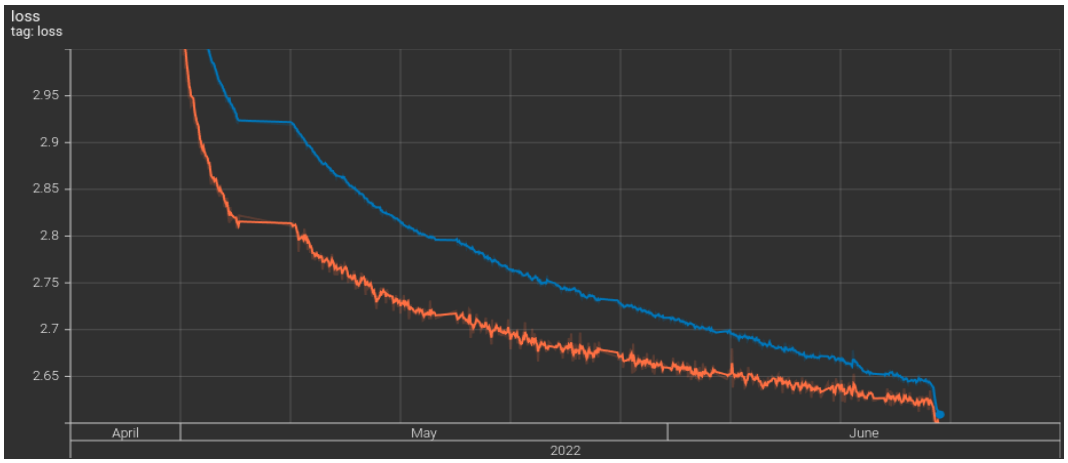


FIGURE C.1: Visualization of **wav2vec 2.0 BASE pre-training loss** movement. The x-axis is the day. The blue line is the training and orange line is the validation.

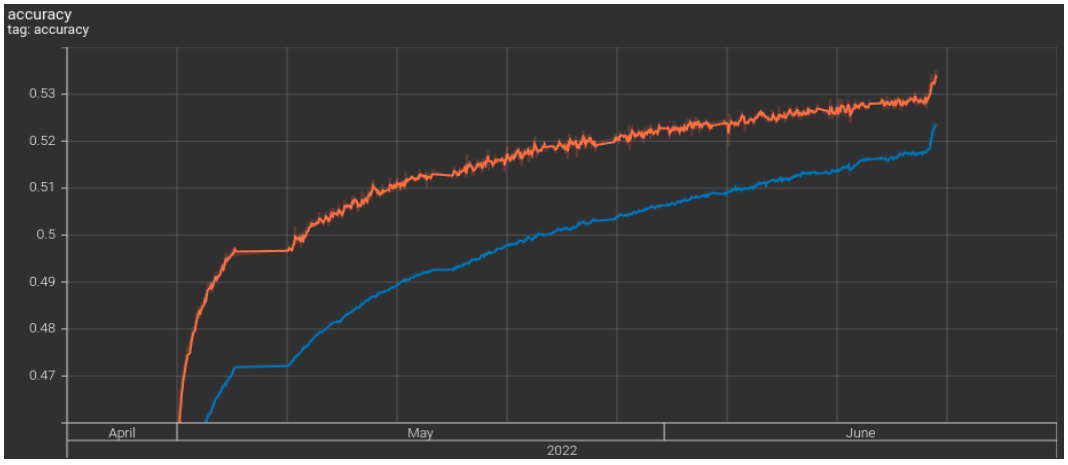


FIGURE C.2: Visualization of **wav2vec 2.0 BASE pre-training accuracy** movement. The x-axis is the day. The blue line is the training and orange line is the validation.

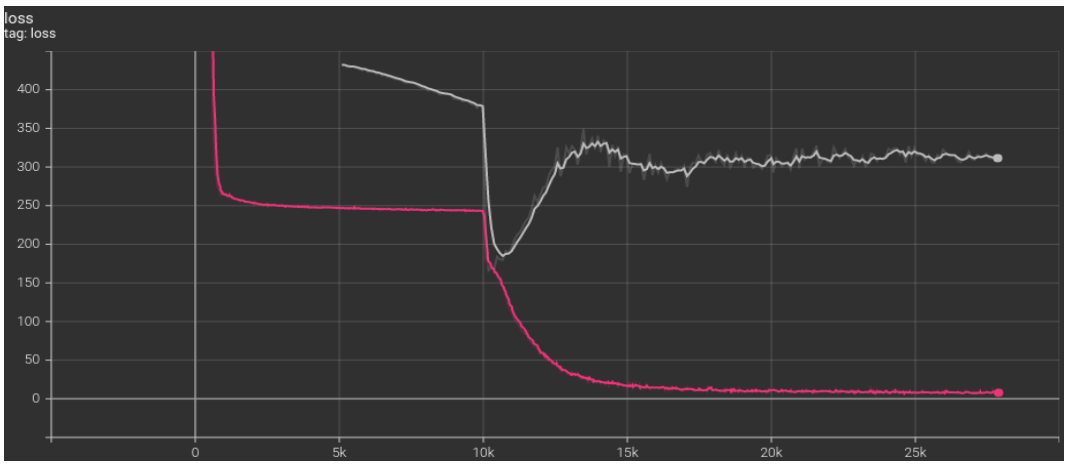


FIGURE C.3: Visualization of **wav2vec 2.0 BASE fine-tuning loss movement without control speakers**. The x-axis is the number of updates. The pink line is the training and white line is the validation. The author has to admit the limitation of the training and possibility for further tuning on hyperparameters as the validation loss did not converge well.

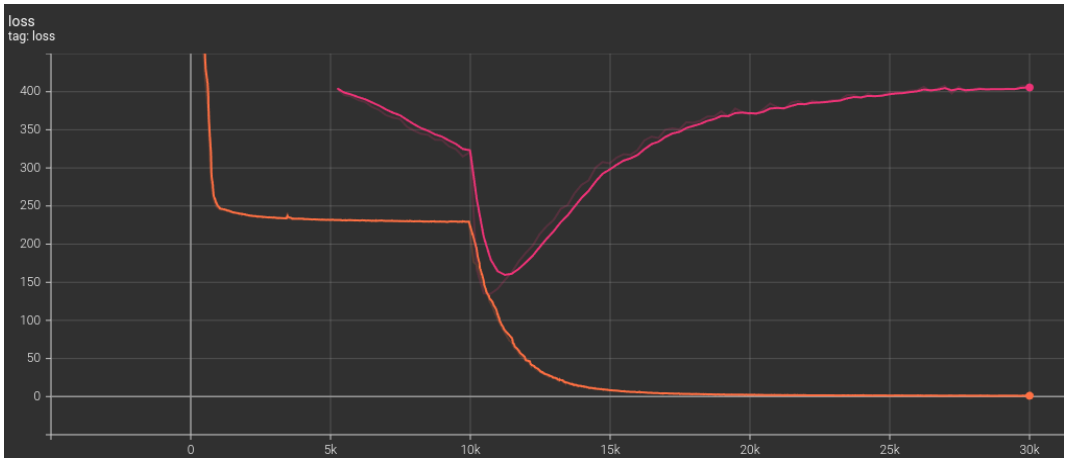


FIGURE C.4: Visualization of **wav2vec 2.0 BASE fine-tuning loss movement with control speakers**. The x-axis is the number of updates. The orange line is the training and pink line is the validation. The author has to admit the limitation of the training and possibility for further tuning on hyperparameters as the validation loss did not converge well.

Appendix D

wav2vec 2.0 XLSR-53 Loss and Accuracy Movement

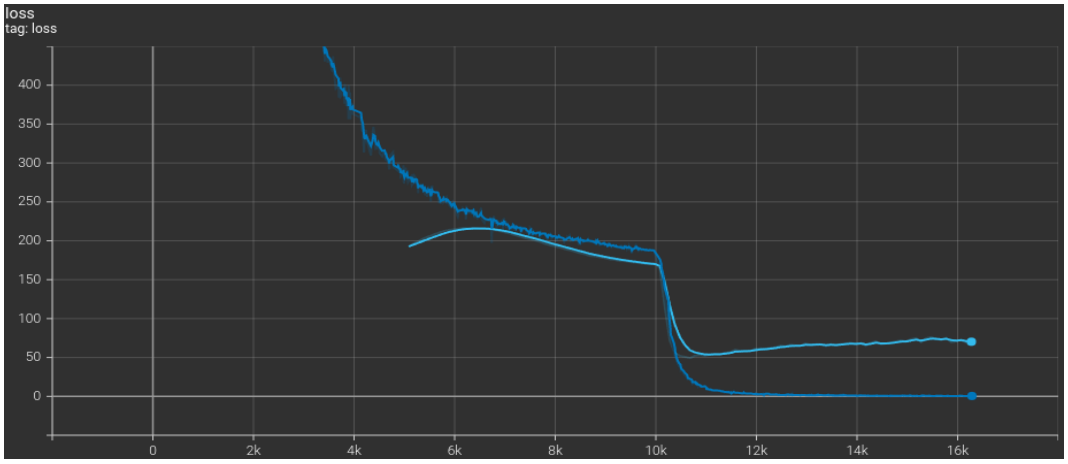


FIGURE D.1: Visualization of **XLSR-53 fine-tuning loss movement without control speakers**. The x-axis is the number of updates. The dark blue line is the training and light blue line is the validation.

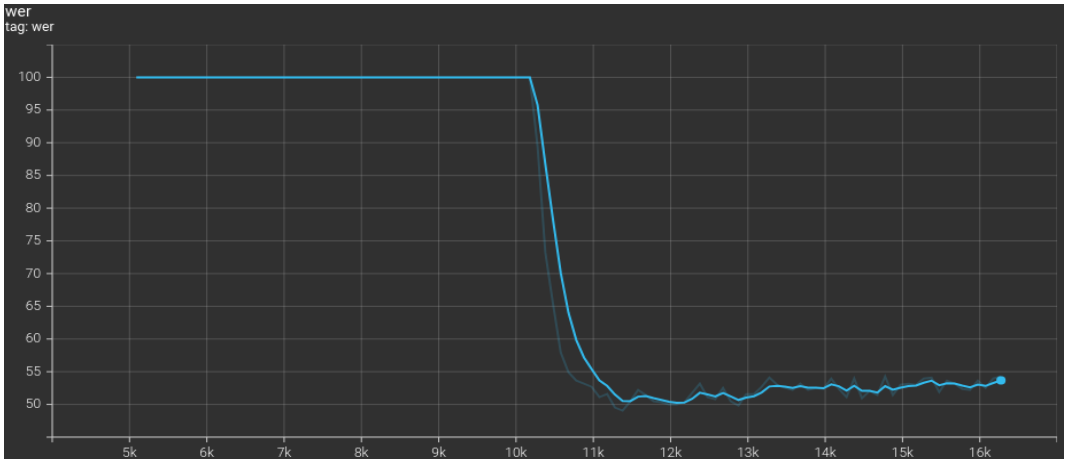


FIGURE D.2: Visualization of **XLSR-53 fine-tuning WER movement without control speakers**. The x-axis is the number of updates. The line is the validation.

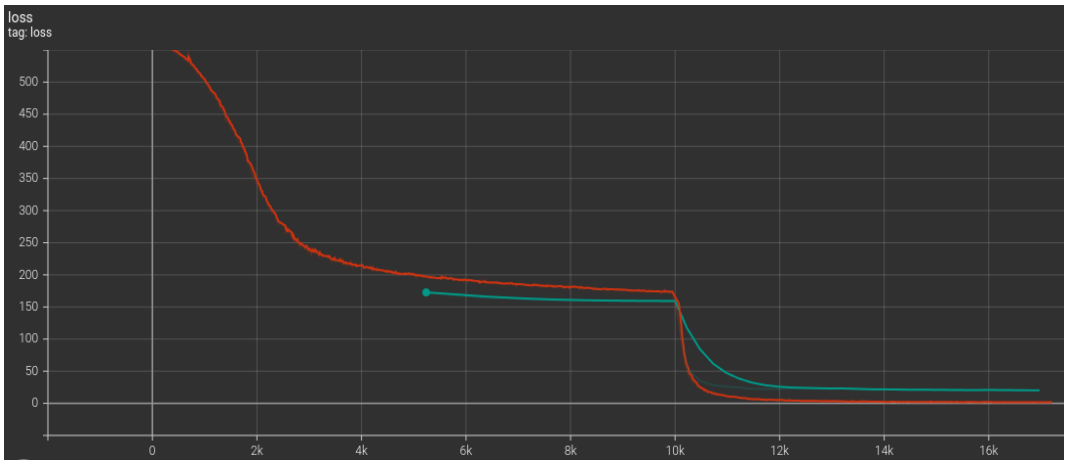


FIGURE D.3: Visualization of **XLSR-53 fine-tuning loss movement with control speakers**. The x-axis is the number of updates. The dark orange line is the training and green line is the validation.

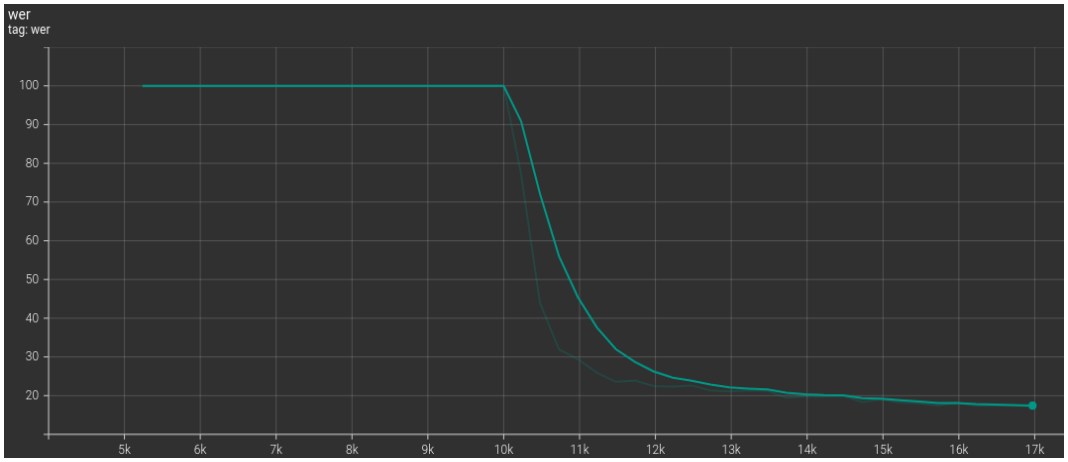


FIGURE D.4: Visualization of **XLSR-53 fine-tuning WER movement with control speakers**. The x-axis is the number of updates. The line is the validation.

Appendix E

wav2vec 2.0 XLSR-53 Re-fine-tuning Loss and Accuracy Movement

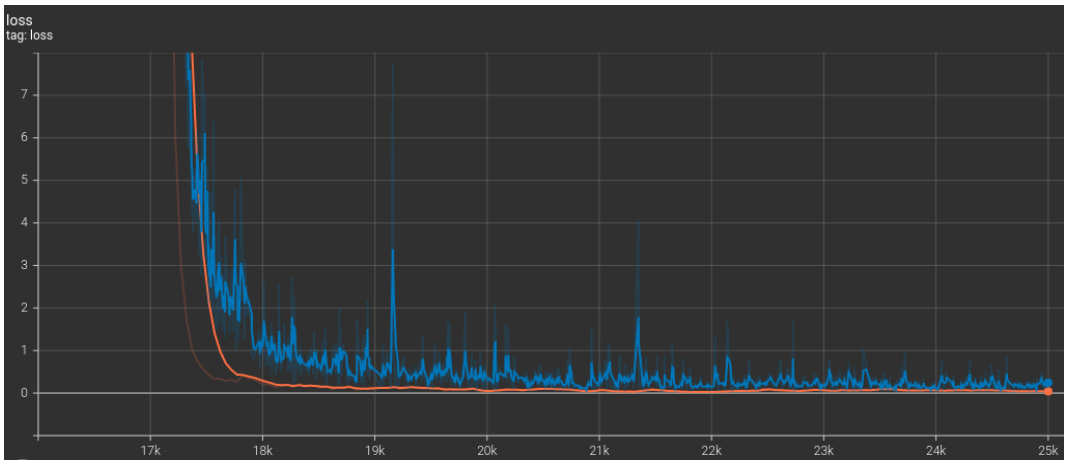


FIGURE E.1: Visualization of **XLSR-53 re-fine-tuning loss movement for target patient ID 17**. The x-axis is the number of updates. The blue line is the training and orange line is the validation.

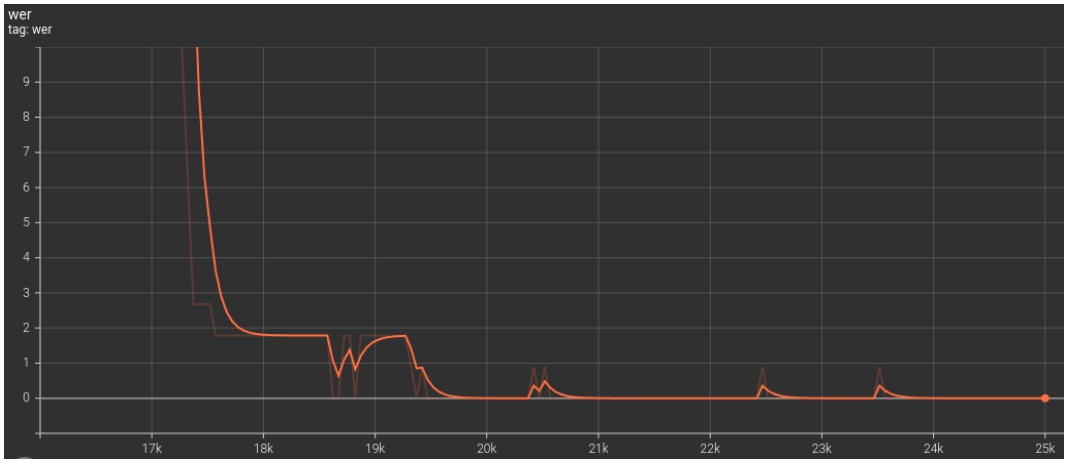


FIGURE E.2: Visualization of **XLSR-53 re-fine-tuning WER movement for target patient ID 17**. The x-axis is the number of updates. The line is the validation.

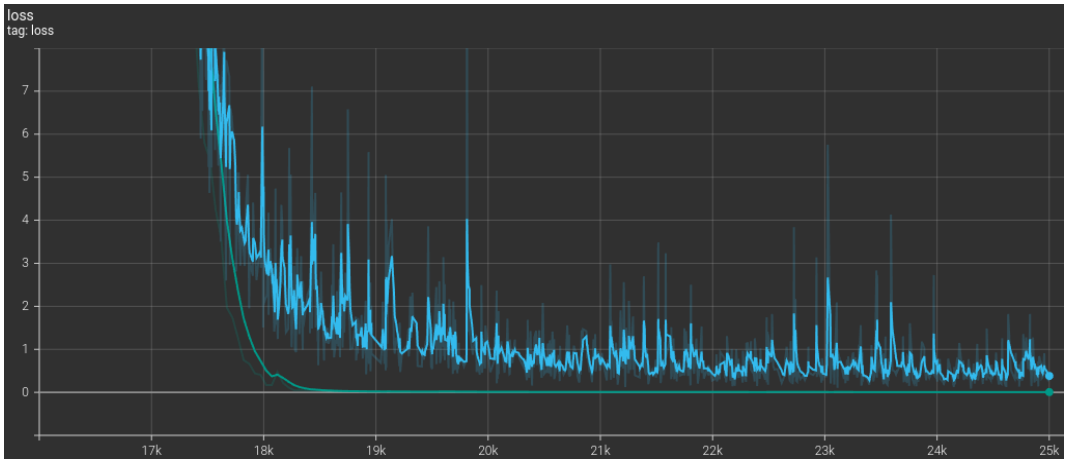


FIGURE E.3: Visualization of **XLSR-53 re-fine-tuning loss movement for target patient ID 28**. The x-axis is the number of updates. The blue line is the training and green line is the validation.

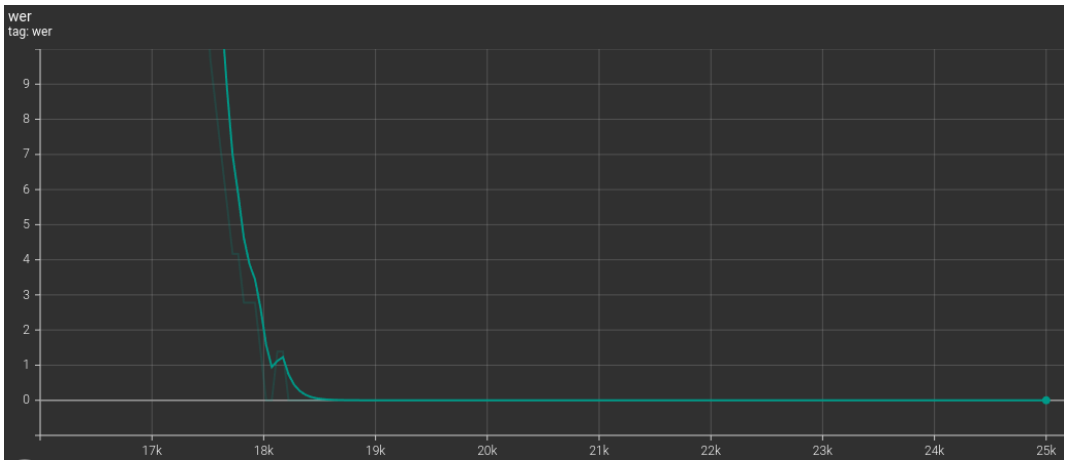


FIGURE E.4: Visualization of **XLSR-53 re-fine-tuning loss movement for target patient ID 28**. The x-axis is the number of updates. The line is the validation.

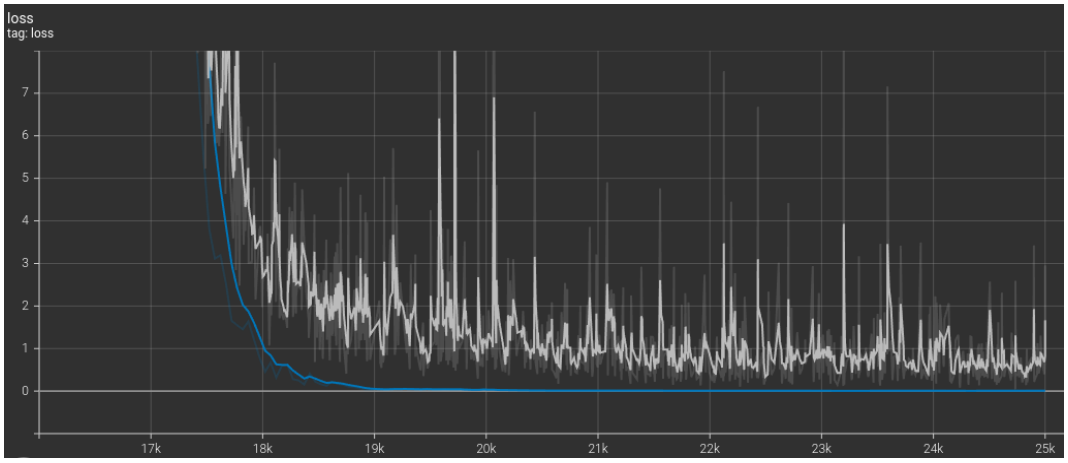


FIGURE E.5: Visualization of **XLSR-53 re-fine-tuning loss movement for target patient ID 41**. The x-axis is the number of updates. The white line is the training and blue line is the validation.

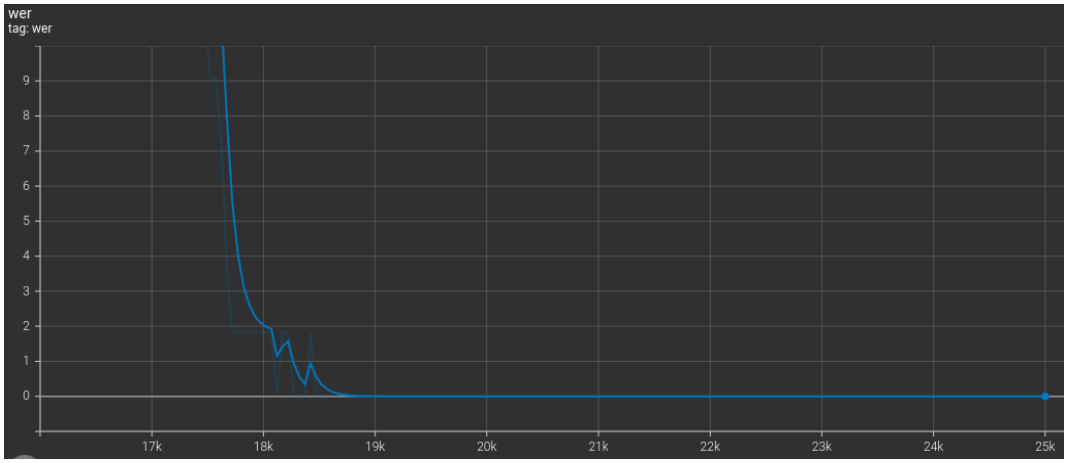


FIGURE E.6: Visualization of **XLSR-53 re-fine-tuning loss movement for target patient ID 41**. The x-axis is the number of updates. The line is the validation.

Bibliography

[1] *Speech recognition*. [Online]. Available: https://paperswithcode.com/task/speech-recognition.

[2] G. Synnaeve, *Syhw/wer$_a$re$_w$e: Attempt at tracking states of the arts and recent results (bibliography) on speech recognition.* [Online]. Available: https://github.com/syhw/wer_are_we.

[3] M. Moore, H. Venkateswara, and S. Panchanathan, "Whistle-blowing asrs: Evaluating the need for more inclusive speech recognition systems," Sep. 2018, pp. 466–470. DOI: 10.21437/Interspeech.2018-2391.

[4] P. Enderby, "Chapter 22 - disorders of communication: Dysarthria," in *Neurological Rehabilitation*, ser. Handbook of Clinical Neurology, M. P. Barnes and D. C. Good, Eds., vol. 110, Elsevier, 2013, pp. 273–281. DOI: https://doi.org/10.1016/B978-0-444-52901-5.00022-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444529015000228.

[5] L. De Russis and F. Corno, "On the impact of dysarthric speech on contemporary asr cloud platforms," *Journal of Reliable Intelligent Environments*, vol. 5, no. 3, pp. 163–172, 2019. DOI: 10.1007/s40860-019-00085-y.

[6] L. P. Violeta, W.-C. Huang, and T. Toda, *Investigating self-supervised pre-training frameworks for pathological speech recognition*, 2022. DOI: 10.48550/ARXIV.2203.15431. [Online]. Available: https://arxiv.org/abs/2203.15431.

[7] B. MacDonald, P.-P. Jiang, J. Cattiau, *et al.*, "Disordered speech data collection: Lessons learned at 1 million utterances from project euphonia," 2021.

[8] H. P. Rowe, S. E. Gutz, M. F. Maffei, K. Tomanek, and J. R. Green, "Characterizing dysarthria diversity for automatic speech recognition: A tutorial from the clinical perspective," *Frontiers in Computer Science*, vol. 4, 2022, ISSN: 2624-9898. DOI: 10.3389/fcomp.2022.770210. [Online]. Available: https://www.frontiersin.org/article/10.3389/fcomp.2022.770210.

[9] F. Xiong, J. Barker, Z. Yue, and H. Christensen, "Source domain data selection for improved transfer learning targeting dysarthric speech recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7424–7428. DOI: 10.1109/ICASSP40776.2020.9054694.

[10] J. R. Green, R. L. MacDonald, P.-P. Jiang, *et al.*, "Automatic Speech Recognition of Disordered Speech: Personalized Models Outperforming Human Listeners on Short Phrases," in *Proc. Interspeech 2021*, 2021, pp. 4778–4782. DOI: 10.21437/Interspeech.2021-1384.

[11] J. Shor, D. Emanuel, O. Lang, *et al.*, "Personalizing asr for dysarthric and accented speech with limited data," Sep. 2019, pp. 784–788. DOI: 10.21437/Interspeech.2019-1427.

[12] M. Geng, X. Xie, S. Liu, *et al.*, "Investigation of data augmentation techniques for disordered speech recognition," in *Interspeech 2020*, ISCA, Oct. 2020. DOI: 10.21437/interspeech.2020-1161. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2020-1161.

[13] B. B. Vachhani, C. Bhat, and S. K. Kopparapu, "Data augmentation using healthy speech for dysarthric speech recognition," in *INTERSPEECH*, 2018.

[14] X. Liu, F. Zhang, Z. Hou, *et al.*, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021. DOI: 10.1109/tkde.2021.3090866. [Online]. Available: https://doi.org/10.1109%5C%2Ftkde.2021.3090866.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2018. DOI: 10.48550/ARXIV.1810.04805. [Online]. Available: https://arxiv.org/abs/1810.04805.

[16] L. Jing and Y. Tian, *Self-supervised visual feature learning with deep neural networks: A survey*, 2019. DOI: 10.48550/ARXIV.1902.06162. [Online]. Available: https://arxiv.org/abs/1902.06162.

[17] A. v. d. Oord, Y. Li, and O. Vinyals, *Representation learning with contrastive predictive coding*, 2018. DOI: 10.48550/ARXIV.1807.03748. [Online]. Available: https://arxiv.org/abs/1807.03748.

[18] S. Schneider, A. Baevski, R. Collobert, and M. Auli, *Wav2vec: Unsupervised pre-training for speech recognition*, 2019. DOI: 10.48550/ARXIV.1904.05862. [Online]. Available: https://arxiv.org/abs/1904.05862.

[19] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, *Wav2vec 2.0: A framework for self-supervised learning of speech representations*, 2020. DOI: 10.48550/ARXIV.2006.11477. [Online]. Available: https://arxiv.org/abs/2006.11477.

[20] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, *Hubert: Self-supervised speech representation learning by masked prediction of hidden units*, 2021. DOI: 10.48550/ARXIV.2106.07447. [Online]. Available: https://arxiv.org/abs/2106.07447.

[21] S. Liu, A. Mallol-Ragolta, E. Parada-Cabeleiro, *et al.*, *Audio self-supervised learning: A survey*, 2022. DOI: 10.48550/ARXIV.2203.01205. [Online]. Available: https://arxiv.org/abs/2203.01205.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[23] K. D. N, P. Wang, and B. Bozza, "Using Large Self-Supervised Models for Low-Resource Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2436–2440. DOI: 10.21437/Interspeech.2021-631.

[24] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, *Applying wav2vec2.0 to speech recognition in various low-resource languages*, 2020. DOI: 10.48550/ARXIV.2012.12121. [Online]. Available: https://arxiv.org/abs/2012.12121.

[25] N. Oostdijk, "The spoken Dutch corpus. overview and first evaluation," in *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece: European Language Resources Association (ELRA), May 2000. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2000/pdf/110.pdf.

[26] J.-P. Martens, M. D. Bodt, G. V. Nuffelen, and C. Middag, "Corpus of pathological and normal speech (copas)," 2011.

[27] E. Yılmaz, M. Ganzeboom, C. Cucchiarini, and H. Strik, "Multi-stage dnn training for automatic recognition of dysarthric speech," Aug. 2017. DOI: 10.21437/Interspeech.2017-303.

[28] B. Ons, J. Gemmeke, and H. Van hamme, "The self-taught vocal interface," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, Dec. 2014. DOI: 10.1186/s13636-014-0043-4.

[29] P. Wang, B. BabaAli, and H. Van hamme, *A study into pre-training strategies for spoken language understanding on dysarthric speech*, 2021. DOI: 10.48550/ARXIV.2106.08313. [Online]. Available: https://arxiv.org/abs/2106.08313.

[30] E. Yılmaz, M. Ganzeboom, C. Cucchiarini, and H. Strik, "Combining Non-Pathological Data of Different Language Varieties to Improve DNN-HMM Performance on Pathological Speech," in *Proc. Interspeech 2016*, 2016, pp. 218–222. DOI: 10.21437/Interspeech.2016-109.

[31] E. Yılmaz, V. Mitra, C. Bartels, and H. Franco, *Articulatory features for asr of pathological speech*, 2018. DOI: 10.48550/ARXIV.1807.10948. [Online]. Available: https://arxiv.org/abs/1807.10948.

[32] E. Yılmaz, V. Mitra, G. Sivaraman, and H. Franco, "Articulatory and bottleneck features for speaker-independent ASR of dysarthric speech," *Computer Speech &amp Language*, vol. 58, pp. 319–334, Nov. 2019. DOI: 10.1016/j.csl.2019.05.002. [Online]. Available: https://doi.org/10.1016%2Fj.csl.2019.05.002.

[33] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2014. DOI: 10.48550/ARXIV.1409.0473. [Online]. Available: https://arxiv.org/abs/1409.0473.

[34] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, *Listen, attend and spell*, 2015. DOI: 10.48550/ARXIV.1508.01211. [Online]. Available: https://arxiv.org/abs/1508.01211.

[35] K. Xu, J. Ba, R. Kiros, *et al.*, *Show, attend and tell: Neural image caption generation with visual attention*, 2015. DOI: 10.48550/ARXIV.1502.03044. [Online]. Available: https://arxiv.org/abs/1502.03044.

[36] G. Letarte, F. Paradis, P. Giguère, and F. Laviolette, "Importance of self-attention for sentiment analysis," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 267–275. DOI: 10.18653/v1/W18-5429. [Online]. Available: https://aclanthology.org/W18-5429.

[37] J. Yu, L. Marujo, J. Jiang, P. Karuturi, and W. Brendel, "Improving multi-label emotion classification via sentiment classification with dual attention transfer network," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 1097–1102. DOI: 10.18653/v1/D18-1137. [Online]. Available: https://aclanthology.org/D18-1137.

[38] L. Wu, F. Tian, L. Zhao, J. Lai, and T.-Y. Liu, "Word attention for sequence to sequence text understanding," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, Feb. 2018. [Online]. Available: https://www.microsoft.com/en-us/research/publication/word-attention-sequence-sequence-text-understanding/.

[39] J. Yang, P. Ren, D. Zhang, *et al.*, *Neural aggregation network for video face recognition*, 2016. DOI: 10.48550/ARXIV.1603.05474. [Online]. Available: https://arxiv.org/abs/1603.05474.

[40] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1347–1360, 2018. DOI: 10.1109/TIP.2017.2778563.

[41] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, 2014. DOI: 10.48550/ARXIV.1409.3215. [Online]. Available: https://arxiv.org/abs/1409.3215.

[42] K. Cho, B. van Merrienboer, C. Gulcehre, *et al.*, *Learning phrase representations using rnn encoder-decoder for statistical machine translation*, 2014. DOI: 10.48550/ARXIV.1406.1078. [Online]. Available: https://arxiv.org/abs/1406.1078.

[43] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, *On the properties of neural machine translation: Encoder-decoder approaches*, 2014. DOI: 10.48550/ARXIV.1409.1259. [Online]. Available: https://arxiv.org/abs/1409.1259.

[44] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017. DOI: 10.48550/ARXIV.1706.03762. [Online]. Available: https://arxiv.org/abs/1706.03762.

[45] A. Gulati, J. Qin, C.-C. Chiu, *et al.*, *Conformer: Convolution-augmented transformer for speech recognition*, 2020. DOI: 10.48550/ARXIV.2005.08100. [Online]. Available: https://arxiv.org/abs/2005.08100.

[46] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2020. DOI: 10.48550/ARXIV.2010.11929. [Online]. Available: https://arxiv.org/abs/2010.11929.

[47] J. Cheng, L. Dong, and M. Lapata, *Long short-term memory-networks for machine reading*, 2016. DOI: 10.48550/ARXIV.1601.06733. [Online]. Available: https://arxiv.org/abs/1601.06733.

[48] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.

[49] M.-T. Luong, H. Pham, and C. D. Manning, *Effective approaches to attention-based neural machine translation*, 2015. DOI: 10.48550/ARXIV.1508.04025. [Online]. Available: https://arxiv.org/abs/1508.04025.

[50] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. DOI: 10.48550/ARXIV.1512.03385. [Online]. Available: https://arxiv.org/abs/1512.03385.

[51] P. Elias, "Predictive coding–i," *IRE Transactions on Information Theory*, vol. 1, no. 1, pp. 16–24, 1955. DOI: 10.1109/TIT.1955.1055126.

[52] B. S. Atal and M. R. Schroeder, "Adaptive predictive coding of speech signals," *Bell System Technical Journal*, vol. 49, no. 8, pp. 1973–1986, 1970. DOI: https://doi.org/10.1002/j.1538-7305.1970.tb04297.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1970.tb04297.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1970.tb04297.x.

[53] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. DOI: 10.48550/ARXIV.1301.3781. [Online]. Available: https://arxiv.org/abs/1301.3781.

[54] C. Doersch, A. Gupta, and A. A. Efros, *Unsupervised visual representation learning by context prediction*, 2015. DOI: 10.48550/ARXIV.1505.05192. [Online]. Available: https://arxiv.org/abs/1505.05192.

[55] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010.

[56] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, *CSR-I (WSJ0) Other*, version DRAFT VERSION, 2016. DOI: 10.7910/DVN/ZVU9HF. [Online]. Available: https://doi.org/10.7910/DVN/ZVU9HF.

[57] P. Woodland, J. Odell, V. Valtchev, and S. Young, "Large vocabulary continuous speech recognition using htk," in *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. ii, 1994, II/125–II/128 vol.2. DOI: 10.1109/ICASSP.1994.389562.

[58] A. Baevski, S. Schneider, and M. Auli, *Vq-wav2vec: Self-supervised learning of discrete speech representations*, 2019. DOI: 10.48550/ARXIV.1910.05453. [Online]. Available: https://arxiv.org/abs/1910.05453.

[59] A. Tjandra, B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, *Vqvae unsupervised unit discovery and multi-scale code2spec inverter for zerospeech challenge 2019*, 2019. DOI: 10.48550/ARXIV.1905.11449. [Online]. Available: https://arxiv.org/abs/1905.11449.

[60] R. Eloff, A. Nortje, B. van Niekerk, *et al.*, *Unsupervised acoustic unit discovery for speech synthesis using discrete latent-variable neural networks*, 2019. DOI: 10.48550/ARXIV.1904.07556. [Online]. Available: https://arxiv.org/abs/1904.07556.

[61] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using WaveNet autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, Dec. 2019. DOI: 10.1109/taslp.2019.2938863. [Online]. Available: https://doi.org/10.1109%2Ftaslp.2019.2938863.

[62] E. J. Gumbel, *Statistical theory of extreme values and some practical applications; a series of lectures*, eng, ser. Applied mathematics series ; 33. Washington: U.S. Govt. Print. Office, 1954.

[63] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, *Unsupervised cross-lingual representation learning for speech recognition*, 2020. DOI: 10.48550/ARXIV.2006.13979. [Online]. Available: https://arxiv.org/abs/2006.13979.

[64] D. Hendrycks and K. Gimpel, *Gaussian error linear units (gelus)*, 2016. DOI: 10.48550/ARXIV.1606.08415. [Online]. Available: https://arxiv.org/abs/1606.08415.

[65] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer normalization*, 2016. DOI: 10.48550/ARXIV.1607.06450. [Online]. Available: https://arxiv.org/abs/1607.06450.

[66] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 369–376, ISBN: 1595933832. DOI: 10.1145/1143844.1143891. [Online]. Available: https://doi.org/10.1145/1143844.1143891.

[67] A. Hannun, "Sequence modeling with ctc," *Distill*, 2017, https://distill.pub/2017/ctc. DOI: 10.23915/distill.00008.

[68] D. S. Park, W. Chan, Y. Zhang, *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*, ISCA, Sep. 2019. DOI: 10.21437/interspeech.2019-2680. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2019-2680.

[69] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Weinberger, *Deep networks with stochastic depth*, 2016. DOI: 10.48550/ARXIV.1603.09382. [Online]. Available: https://arxiv.org/abs/1603.09382.

[70] A. Fan, E. Grave, and A. Joulin, *Reducing transformer depth on demand with structured dropout*, 2019. DOI: 10.48550/ARXIV.1909.11556. [Online]. Available: https://arxiv.org/abs/1909.11556.

[71] R. Ardila, M. Branson, K. Davis, *et al.*, *Common voice: A massively-multilingual speech corpus*, 2019. DOI: 10.48550/ARXIV.1912.06670. [Online]. Available: https://arxiv.org/abs/1912.06670.

[72] M. Gales, K. Knill, A. Ragni, and S. Rath, *Speech recognition and keyword spotting for low-resource languages : Babel project research at cued*, © 2014 ISCA. Reproduced in accordance with the publisher's self-archiving policy., May 2014. [Online]. Available: https://eprints.whiterose.ac.uk/152840/.

[73] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "MLS: A large-scale multilingual dataset for speech research," in *Interspeech 2020*, ISCA, Oct. 2020. DOI: 10.21437/interspeech.2020-2826. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2020-2826.

[74] G. Hinton, L. Deng, D. Yu, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012. DOI: 10.1109/MSP.2012.2205597.

[75] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," 2017. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/pdfs/0233.PDF.

[76] L. P. Violeta, W.-C. Huang, and T. Toda, *Investigating self-supervised pre-training frameworks for pathological speech recognition*, 2022. DOI: 10.48550/ARXIV.2203.15431. [Online]. Available: https://arxiv.org/abs/2203.15431.

[77] A. Hernandez, P. A. Pérez-Toro, E. Nöth, J. R. Orozco-Arroyave, A. Maier, and S. H. Yang, *Cross-lingual self-supervised speech representations for improved dysarthric speech recognition*, 2022. DOI: 10.48550/ARXIV.2204.01670. [Online]. Available: https://arxiv.org/abs/2204.01670.

[78] L. Wu, D. Zong, S. Sun, and J. Zhao, "A sequential contrastive learning framework for robust dysarthric speech recognition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7303–7307. DOI: 10.1109/ICASSP39728.2021.9415017.

[79] S. Chen, C. Wang, Z. Chen, *et al.*, *Wavlm: Large-scale self-supervised pre-training for full stack speech processing*, 2021. DOI: 10.48550/ARXIV.2110.13900. [Online]. Available: https://arxiv.org/abs/2110.13900.

[80] C. Middag, j.-p. Martens, G. Nuffelen, and M. Bodt, "Dia: A tool for objective intelligibility assessment of pathological speech," Sep. 2009.

[81] D. Povey, G. Cheng, Y. Wang, *et al.*, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in *Proc. Interspeech 2018*, 2018, pp. 3743–3747. DOI: 10.21437/Interspeech.2018-1417.

[82] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014. DOI: 10.48550/ARXIV.1412.6980. [Online]. Available: https://arxiv.org/abs/1412.6980.