Automatic Dysarthria Severity Assessment using Whisper-extracted Features

Evaluating ML architectures for dysarthria severity assessment on TORGO and MSDM

Christopher Charlesworth[1]
Supervisor(s): Zhengjun Yue[1], YuanYuan Zhang[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Christopher Charlesworth
Final project course: CSE3000 Research Project
Thesis committee: Zhengjun Yue, YuanYuan Zhang, Thomas Durieux

Abstract

Dysarthria is a speech disorder commonly caused by neurological disorders such as strokes, cerebral palsy and Amyotrophic Lateral Sclerosis (ALS). The severity level of dysarthria greatly influences the appropriate treatment for a patient. However, assessing the severity of dysarthria in a patient is a time-consuming process that requires a trained speech therapist. Therefore the following work explores a variety of classifier architectures for automatic dysarthria severity assessment using Whisper encodings. The datasets used were MSDM and TORGO while the classifier architectures implemented included a Convolutional Neural Networks and Recurrent Neural Network variants. Across both datasets, the Gated Recurrent Unit network (GRU) achieved the best performance with 97.21% accuracy on MSDM and 97.47% on TORGO.

Index Terms: speech recognition, dysarthria

1. Introduction

1.1. Background

Dysarthria is a speech disorder caused by weakness or incoordination of the muscles necessary for speech. It is commonly caused by strokes, Parkinson's disease, cerebral palsy and other neurological disorders [1]. With close to 90% of individuals with Parkinson's disease having dysarthria, we see that it is a common and serious issue that requires great attention [2]. Dysarthria can cause people's speech to be unintelligible which has been shown to impact their self-esteem, ability to express themselves and their overall autonomy [3]. Luckily, speech and language therapy can help patients improve their speech intelligibility, but the focus of this therapy is highly dependent on the severity level of the patient's dysarthria [3]. Today, the most common form of dysarthria assessment is via a test, such as the Frenchay Dysarthria assessment [4]. However, these tests are time-consuming and need to be performed by a licensed speech therapist. Therefore, there is a need for automatic dysarthria severity assessment based on recordings of patients' speech.

1.2. Related Work

Previous work done in this area includes Joshy et al. comparing the performance of different classifier architectures for dysarthria severity assessment [5]. The models considered included a Support Vector Machine (SVM), Random Forest (RF), Deep Neural Network (DNN), Convolutional Neural Network (CNN), Long Short-Term Memory network (LSTM) and Gated Recurrent Unit network (GRU). The models were evaluated when trained on different spectral features namely Mel-Frequency Cepstral Coefficients (MFCCs) and Constant-Q Cepstral Coefficients (CQCCs). Moreover, the models were trained and evaluated on different datasets, specifically TORGO [6] and UASpeech [7]. Their highest reported accuracy was 96.18% which was achieved by training a CNN on MFCCs. Furthermore, their GRU outperformed all models on both TORGO and UASpeech when trained on CQCC features. Additionally, their models achieved higher levels of accuracy when trained on MFCCs compared to CQCCs and all models performed better when trained on TORGO compared to UASpeech. To assess speaker dependency, the authors also performed Leave One Speaker Out (LOSO) cross-validation allowing them to measure how the models perform on new speakers not included in the training data. However, due to the imbalanced nature of UASpeech the models were adapted to perform binary dysarthria detection rather than dysarthria severity assessment. This means their conclusion that models trained on CQCCs perform better on new speakers than those trained on MFCCs may only hold for dysarthria detection.

Next, Charola et al. developed a dysarthria severity classifier using the encoding of Whisper's model followed by a CNN [8]. Their focus was on evaluating the effect of noisy conditions on the performance of their model. Their best-performing CNN achieved an accuracy of 97.49% with this being more than a 1% accuracy improvement compared to state-of-the-art models trained on MFCCs. Additionally, models trained on Whisper features were shown to outperform models trained on MFCCs by over 10% in accuracy when noise was added to the utterances. However, the study only used 1,982 utterances from TORGO, with this being a small subset of the entire TORGO dataset [6]. Additionally, this subset only included 6 of the 7 speakers with dysarthria included in TORGO. Finally, the results produced were from one split of 90% training data and 10% test data with no repeated testing conducted. This means that the results could have been produced by a 'lucky' testing and training split and do not represent the true performance of their model.

Mani et al. developed a dysarthria detection classifier by applying a CNN to the Mel Spectrogram of the given unit of speech [9]. Furthermore, they applied transfer learning by fine-tuning the ResNet50 CNN, a popular image classification model to perform dysarthria detection. Their best solution reached an impressive level of accuracy with 97.73% on the TORGO corpus. Additionally, they found that their fine-tuned ResNet50 model performed significantly better than the CNN that they trained from scratch. Although these results are promising, they are comparing the performance of different models on only one dataset, namely TORGO. To determine if their transfer-learned CNN truly outperforms a CNN trained from scratch for dysarthria detection, results from more than one dataset would be needed.

In another paper, Rathod et al. compared the performance of CNNs trained on Whisper embeddings with those trained on traditional sound representations, such as MFCCs and Linear Frequency Cepstral Coefficients (LFCCs), for dysarthria severity assessment [10]. The CNNs developed included one trained from scratch as well as a ResNet50 model that they fine-tuned in a similar manner as the authors in [9]. The models were trained and evaluated on the TORGO corpus as well as UASpeech [6] [7]. Both the newly trained CNN and fine-tuned ResNet50 achieved their best performance when trained on TORGO with accuracies of 98.49 % and 98.99 % respectively. Additionally, their results showed that their models achieved higher levels of accuracy when trained on Whisper embeddings, compared to both MFCCs and LFCCs. One critique of the paper is that they only compared the performance of two CNNs and did not consider other families of classifiers such as those developed in [5]. Additionally, the results presented are achieved by selecting the highest performance of the models across 5 runs rather than averaging the performance of these runs.

1.3. Proposed Work

The proposed solution of this paper is to adapt Whisper, a traditional Automatic Speech Recognition (ASR) model into a dysarthria severity classifier. Whisper has an encoder-decoder architecture and is a weakly-supervised model, trained on over 680,000 hours of audio from across the internet [11]. Whisper's encodings have been shown to robustly and efficiently repre-

sent the spectral information of sound [8]. In addition to this, dysarthria severity classifiers trained on Whisper embeddings have been shown to achieve greater levels of accuracy compared to traditional spectral representations like MFCCs and LFCCs [10]. However, to the best of our knowledge, the types of classifiers trained on Whisper embeddings for dysarthria severity assessment have been limited to variations of CNNs. In contrast, other machine learning techniques such as GRUs have been shown to produce impressive results when trained on traditional spectral representations [5]. Hence, the proposed work explores the performance of both a CNN and RNN variations for dysarthria severity assessment using Whisper-extracted features.

The main research questions that this paper aims to answer include:

- How do different types of classifiers perform in distinguishing between dysarthria severity levels using Whisper's encodings?
- How do training classifiers on different dysarthria datasets impact their performance?
- How does the inclusion of padded silence in the Whisper embeddings affect the performance of the classifiers?
- How does fine-tuning Whisper to perform dysarthric ASR affect the performance of classifiers trained on its encodings?

The types of classifiers that have been considered are a CNNs, LTSMs, Bidirectional Long Short-Term Memory network (BiLSTM), GRUs and a traditional RNN. The datasets the models are trained on include TORGO [6] and MSDM [12]. Furthermore, the evaluation metrics include accuracy, F1 score, Jaccard score and Matthew's Correlation Coefficient (MCC).

The main conclusions include that the best-performing model was the GRU, with it reaching an impressive accuracy of 97.11% on the TORGO dataset. This was further improved to 97.47% when trained on fine-tuned Whisper embeddings that were processed to include no embedded silence.

The remainder of this paper includes six main sections: Section 2 discusses the methodology, which describes the selected model's architectures and provides more details on the end-to-end training pipeline. Section 3 describes the datasets that are used as well as an in-depth explanation of the experiment setup. Section 4, Results, shows the performance of the respective models on each dataset. Section 5 reflects on the results achieved and the limitations of the research. Section 6 summarizes the main research questions and findings and proposes possible future work. Finally, Section 7 discusses the reproducibility of the results and the ethical considerations made regarding data storage.

2. Methodology

2.1. Whisper Model

Open AI's Whisper is currently considered to be one of the state-of-the-art ASR models with it being shown to outperform its main point of comparison Wav2vec2 on key metrics such as Word Error Rate (WER) [11]. One of the main distinctions between Whisper and its contemporaries is that it is a weakly supervised model, meaning a portion of its data is labeled with the rest unlabeled [11]. The incorporation of large amounts of unlabeled data from the internet has likely contributed to the robustness of Whisper's embeddings [8]. Furthermore, Whisper is also a multilingual ASR model with it being trained on data from 97 languages, more specifically, of its 680,000 hours of training audio, 117,000 hours are from languages other than English. In addition to being multilingual, Whisper is also multitask. Multitask models leverage shared information and representations across multiple tasks to improve both their performance and efficiency [11]. Whisper's multi-lingual and multitask properties have allowed its training data to have greater diversity than a traditional ASR model. This training data diversity has enabled Whisper to learn rich, generalized speech embeddings, which is beneficial for nuanced tasks like dysarthria assessment [8].

The Whisper model has a variety of sizes, namely Tiny, Base, Small, Medium and Large. They differ in terms of the number of parameters considered, the number of layers and importantly for this topic, the width of their encodings [11]. Rathod et al. found that the large model performed best for training their CNN to assess severity levels of dysarthria using Whisper's encodings [8]. Therefore, the large model was selected for our training pipeline.

2.2. Training Pipeline

Figure 1 shows the end-to-end training pipeline of all models trained on Whisper encodings. Firstly, the sound is converted into an 80-channel log-Mel spectrogram. This is then fed into 2 convolutional layers with a kernel size of 3. Next, sinusodial embeddings are calculated to help Whisper learn the relative positions within the input signal. This, along with the output of the convolutional layers, are fed into a fixed number of encoder blocks whose output produces a shape of 1280×1500 [11]. A width of 1280 is due to the large Whisper model being selected, while the timeframe of 1500 is due to Whisper padding silence to all utterances. More specifically, Whisper pads all input audio to 30 seconds and given that the encodings represent 50ms slices of the input sound, the total length reaches 1500 units. However, since both MSDM and TORGO have utterances that are much shorter than 30 seconds, Whisper pads large amounts of silence to the end of each utterance. This excess padding led to worse performance (see section 4) and was less memory efficient since large amounts of embedded silence was stored for each utterance. To combat this, the timeframes of the embeddings were cut to either 375 or 125 for TORGO and 40 for MSDM since its utterances are shorter (see section 3.1.2). Figure 1 shows an example of this with the embeddings cut to a timeframe of 375. The classifier is either the CNN, RNN, LSTM, GRU or BiLSTM, all of which were developed using Pytorch.

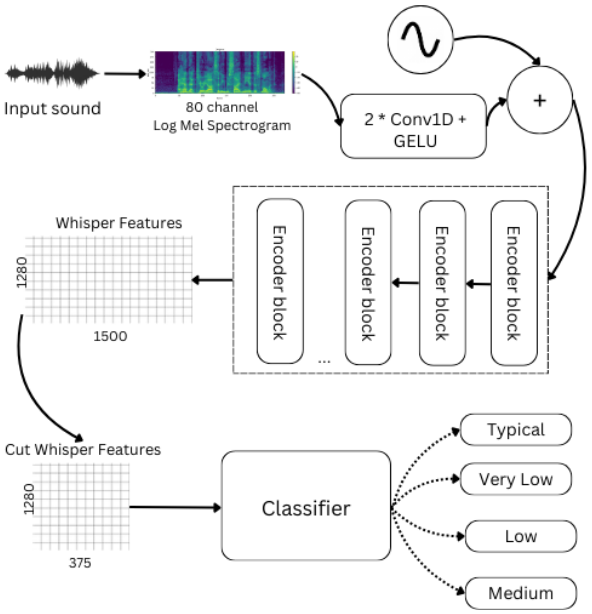

Figure 1: *Training pipeline for all models with Whisper embeddings cut to a timeframe of 375*

2.3. Fine-tuned Whisper model

Fine-tuning is a deep learning technique that involves taking a pre-trained model (which has been trained on large amounts of data) and updating its weights by further training it on a smaller dataset. The result is a model that leverages the knowledge gained in the original training to perform better than a model trained only on the smaller dataset [13]. To assess how fine-tuning Whisper for dysarthric ASR affects the performance of classifiers trained on its embeddings, the MG Whisper model was selected. The MG Whisper model was developed by Mirella Günther for their Bachelor's research project by training the Whisper large-V3 model on TORGO for 2 epochs. The weights were updated using low-rank adaptation (LoRa), a parameter-efficient technique that reduces the number of trainable parameters compared to full fine-tuning [14]. After training, the MG Whisper model improved the WER by 20.49% for dysarthric speakers with a Low dysarthria severity level compared to the unchanged Whisper model.

2.4. Model Architectures

2.4.1. CNN Architecture

As previously stated, this paper explores a variety of classifier architectures, including a CNN, traditional RNN, LSTM, BiLSTM and GRU. Firstly, CNN is a type of deep learning algorithm designed for handling 2D data, with its most common application being image classification [15]. In this use case, one could imagine the sound as an image as it also has two dimensions, the first being time and the second being the embeddings calculated for this time interval. As Figure 2 shows, CNNs utilize convolutional layers to learn spatial features compared to traditional ANNs which do not account for the spatial structure of input data. The results of the convolutional layers are fed into pooling layers, with the most common being a max-pooling layer. One benefit of utilizing a CNN for dysarthria severity assessment is its hierarchical feature learning which allows it to learn complex patterns by combining low-level learned features. Furthermore, CNNs are translationally invariant, meaning they can recognize learned features regardless of their position in the input data [15].

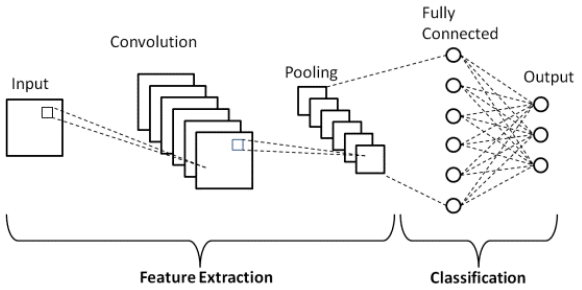

Figure 2: *Example of a CNN Architecture [16]*

The specific CNN architecture chosen for TORGO was four convolutional layers with the first two having a kernel size of 3 and the final two having a kernel size of 5. The number of output layers each convolutional layer produced was 16, 32, 64 and 128 respectively. The CNN architecture for MSDM was altered since the input sizes differed significantly between the datasets. For MSDM, three convolutional layers were applied with each having a kernel size of 3 and their output layers being 8, 16, and 32. These output layers were then passed through a ReLU activation function, as is typical for CNNs [15]. Next, a max-pooling layer with a kernel size of 2 was applied to each layer except for the final layer, which was flattened before being fed into a 1-layer deep linear model. Finally, the CNN, as with all other models discussed in this paper, used cross entropy loss and stochastic gradient descent to learn the optimal weights.

2.4.2. RNN Variants

Recurrent Neural Networks (RNNs) are a type of neural network that detects and exploits patterns in sequential data. Common applications for RNNs include predicting stock prices and natural language processing tasks for example text generation [17]. RNNs differ from traditional artificial neural networks significantly due to how information is propagated through the network. RNNs compute their output by incorporating both the current input data and the model's hidden state from the previous time step, thus incorporating the sequential nature of the data [17]. Despite their advantages in handling sequential data, traditional RNNs struggle to learn long-term dependencies due to their vanishing and exploding gradients in training. The vanishing gradient problem occurs because the gradient, which measures how the current state is influenced by some earlier state, is repeatedly multiplied by itself for each time step between the two states. Hence, if this gradient is less than 1, this repeated multiplication will trend to 0 given enough time steps. The exploding gradient is the same issue, except if the gradient is greater than 1, the repeated multiplication will make it reach infinity [17].

Luckily, there are types of RNNs that do not suffer from the exploding or vanishing gradient problems with one of the most common being LSTMs. LSTMs maintain long-term dependencies through the use of a cell state, which can be seen as a form of memory. The goal is to store information in the cell state that is relevant for calculating future outputs [18]. To do so, LSTMs make use of 3 gates: the forget gate, input gate and

output gate. Each gate can be thought of as a learned function that decides how much of its input to let through, their output is passed through a sigmoid function to ensure it is in the range of 0 to 1. The forget gate is responsible for which portion of the cell to forget. The output gate decides what part of the cell state should contribute to the current output. Finally, the input gate determines what new information should we add or update to the cell state [18]. The main benefit of utilizing LSTMs in the field of dysarthria severity assessment is that they do not suffer from exploding or vanishing gradient problems, meaning they can form longer temporal dependencies compared to traditional RNNs. Additionally, their incorporation of gates allows for a more controlled flow of information from cell to cell compared to traditional RNNs [18]. BiLSTMs can be considered an extension of LSTMs with one key addition. They process the input in both forward and backward directions, allowing them to capture greater contextual information in the cell state [19].

The final type of RNN considered in this research is Gated Recurrent Units (GRUs). GRUs are similar to LSTMs but they differ in their number of gates and their efficiency [20]. GRUs have only two gates: the update gate and the reset gate. The update gate is responsible for determining what of the past cell state should be kept as well as how much of the input should be included in the new cell state. The reset gate determines how much of the past information to forget as it is deemed not relevant to the current time step. Regarding efficiency, GRUs are 29.29% faster at processing the same dataset compared to LSTMs, this difference is even more significant when comparing GRUs with BiLSTMs as they need to process the inputs in both directions [20]. In the field of dysarthria assessment, GRUs have been shown to achieve higher accuracies than other RNN variants such as LSTMs [5]. This can be attributed to their simplified gate structure being less prone to overfitting on training data compared to LSTMs [20].

3. Experiment

3.1. Datasets

3.1.1. Torgo Dataset

All models were trained on a subset of the TORGO database [6]. TORGO contains utterances from 15 speakers, 8 of whom have dysarthria with either ALS or cerbral palsy and 7 of whom have typical speech. Table 1 shows the dysarthria severity of the participants whose recordings were used to train the models. TORGO consists of a variety of utterances including short words like "yes", restricted sentences, unrestricted sentences and non-words [6]. Restricted sentences in this case were phonetically rich sentences that the participants were tasked with speaking aloud. Unrestricted sentences were gathered by having the participants describe a variety of images in whichever way they saw fit. Finally, non-words were repetitions of syllables that were again phonetically rich [6].

The subset of TORGO selected for this study consisted of 3667 utterances lasting longer than 2.5 seconds. These included 867 labeled with a dysarthria severity level of Very Low, 923 Low, 927 Medium and 950 Healthy. By ensuring that all labels had roughly the same number of data points, we avoided our model overfitting to the distribution of our training set.

Severity	**Participant**
Typical	MC01
	FC01
	MC04
VeryLow	F04
	M03
Low	M05
	F03
Medium	F01
	M01
	M02
	M04

Table 1: *TORGO Participant Dysarthria Severity [21]*

3.1.2. MSDM Dataset

In addition to TORGO, each model was trained and evaluated on the MSDM dataset, which is a dysarthria dataset in Mandarin. More specially, MSDM includes 25 participants with dysarthria who had a stroke and 25 typical speakers [12]. Participants were tasked with uttering 200 unique syllables, 90 common characters, 150 common words and 72 sentences. For this study, a total of 61,396 unique utterances were considered, of which 22,753 belonged to label 0, 23074 to label 1, 10,613 to label 2 and 4956 to label 3. To balance the distribution of the dataset, the minority classes were randomly sampled to reach 23,074 samples.

3.2. Model Training and Optimization Techniques

The datasets were split such that 90% of the data was allocated for training and 10% for testing. To achieve more accurate results repeated testing was applied such that the models were trained 10 times each on a different test and train split with the evaluation metrics then averaged. Additionally, the training set was further split into 90% for true training, while 10% was allocated as a validation set. This validation set was used to determine when to perform early stopping if the model's performance was no longer improving. More specifically, patience-based early stopping was implemented such that if the average loss on the validation set had not improved after 13 epochs, then the training would terminate and the best-performing version would be evaluated on the test set. Figure 3 shows an example of this early stopping in effect for a training of the GRU on the TORGO dataset. We see that the validation loss, validation accuracy and test accuracy improve quickly in the first epochs with the test accuracy in this run plateauing after 28 epochs. The inclusion of early stopping prevents the model from overfitting on the training data supplied and also is more computationally efficient. The learning rate of the models was adjusted in a similar manner, if the performance on the validation set had not improved after 3 epochs, then the learning rate would be multiplied by 0.5 with the initial learning being set to 0.1 and the minimum learning rate set to 0.00001.

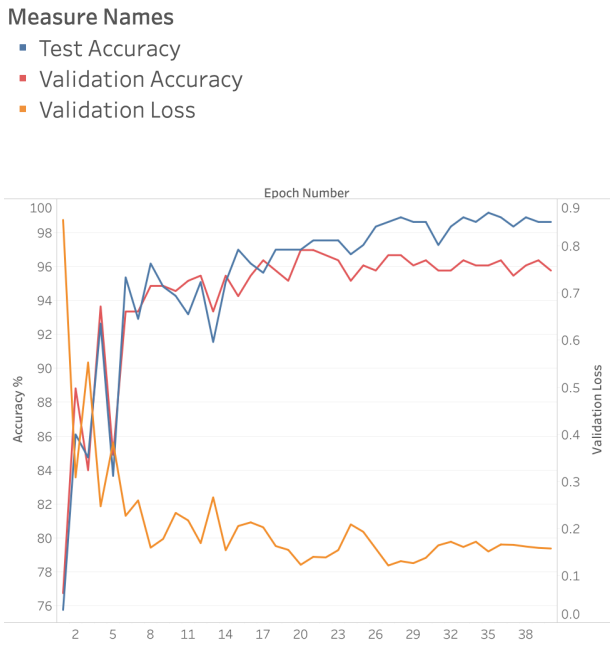

Figure 3: *Training accuracy, validation accuracy and validation loss for a GRU run on TORGO embeddings cut to a timeframe of 125*

3.3. Experiment Configurations

All models were run on both TORGO and MSDM to assess the effect of different datasets on the performance of the models. Additionally, the models were run on TORGO with the embeddings cut to a timeframe of 375 and 125 to assess the impact of padded silence on the performance of the models. Finally, to evaluate the effect of fine-tuning Whisper for dysarthric speech, all models were trained on fine-tuned Whisper embeddings of the TORGO dataset with no padded silence included. The specific architectures implemented for the RNNs include a hidden state size of 128 except for the BiLSTM which has a hidden state size of 256 since it needs to store information about both directions of the input. Furthermore, the networks themselves were 2 layers deep and all models were trained using stochastic gradient descent and cross-entropy loss.

3.4. Evaluation Metrics

The models were evaluated on the following metrics: accuracy, F1 score, Jaccard score and Mathew's Correlation Coefficient (MCC). Accuracy, of course, is the number of correctly labeled samples divided by the total number of samples. The formula below provides more information with TP representing true positive, FN representing false positives etc. If one were to only evaluate using accuracy, you might not consider some aspects of the performance of the model. For example, a model may reach relatively high levels of accuracy by predicting the majority class for every data point.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Metrics like F1 score, Jaccard score and MCC aim to provide a more nuanced evaluation of a model's performance. An F1 score is calculated by taking the harmonic mean of the model's precision and recall. Precision is the number of true positives divided by the total number of samples predicted to be positive. Conversely, recall is the number of true positives divided by the total number of data points labeled as positive. It is important to note that F1 scores can only be calculated for binary classification, to account for this the macro average of F1 scores was taken.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1 Score} = 2 \times \frac{precision \times recall}{precision + recall}$$

The Jaccard score measures the similarity of two sets by taking the intersection of the sets divided by the union of them. Once again, the macro average was taken by performing a one-vs-rest approach.

$$\text{Jaccard Score} = \frac{TP}{TP + FP + FN}$$

Finally, Matthew's Correlation Coefficient (MCC) shows the degree of relation between the expected and actual class. It has a range of - 1 to 1 with 0 being randomly assigning classes, 1 being perfect predictions and -1 being when no data points are predicted correctly.

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

4. Results

4.1. TORGO Results

As mentioned in section 3.1, the TROGO subset was selected by taking all utterances longer than 2.5 seconds. With almost no recordings lasting longer than 7.5 seconds, the embeddings were cut to this point ie. to a timeframe of 375 instead of 1500. The results for all of these models can be seen in Table 2. Additionally, the models were also evaluated on the case when the embeddings were cut to be 2.5 seconds long, essentially ensuring that there was no silence at the end of any embedding. The results of these models can be found in Table 3.

We see in Table 2 that the CNN and LSTM have achieved similar performance with accuracies of 96.21% and 95.32%, respectively. Furthermore, with both having very high MCC scores, we see that the models have truly learned the features and are not relying on random guessing. The LSTM has achieved this strong performance by learning long-term dependencies and patterns in the temporal sequence by utilizing its memory cells to retain important information. The CNN, alternatively, has leveraged its convolutional layers to extract local patterns and features from the speech signals. Next, the GRU had the best performance on all of the metrics. With an accuracy of 97.18% and a high F1, Jaccard Score and MCC the model is achieving high performance on all classes. Furthermore, the GRU has outperformed the LSTM, this is consistent with previous research that shows GRUs outperform LSTMs on small datasets or ones with low-complexity sequences [20]. This is because their simplified gate architecture is less prone to overfitting on the training data. Moreover, we see that the traditional RNN performed worst out of all of the models, only achieving an accuracy of 31.92% when the dataset was cut to a length of 375. Additionally, we see that its MCC score was

0.1173, indicating that the model only barely outperformed random guessing. This can be attributed to the vanishing and exploding gradient problems that inhibit traditional RNNs from learning long-term dependencies as well as GRUs or LSTMs, see Section 2.4.2 for more information.

Table 3 shows that the performance of all RNNs improved when trained on data that was cut such that there was no silence embedded. This effect was most visible with the traditional RNN, with its accuracy improving by 27.04%, one could contribute this to the vanishing or exploding gradient problem. Given that the embeddings cut to 7.5 seconds had up to 5 seconds of embedded silence padded to them, removing this silence likely helped mitigate the gradient issues by reducing the length of the sequence and reducing the amount of data that does not contribute to learning the patterns. The BiLSTM also improved much more significantly than the LSTM or GRU when processing embeddings that had been cut so that there was no silence. More specifically, its accuracy improved by 11.62% while the GRU and LSTM improved by 0.68% and 0.07% respectively. Unlike the traditional RNN, this improvement is not attributable to the vanishing gradient problem as the BiLSTM does not suffer from it. Instead, because the BiLSTM processes data in both the forward and backward directions, the backward direction starts with up to 5 seconds of padded silence, meaning its contribution is negatively impacting the model's performance. Finally, we saw that the CNN's performance did not improve when trained on embeddings cut so that there was no silence padded. This can be attributable to the feature extraction method of CNNs, which focus on local patterns rather than temporal sequence meaning they are less impacted by the presence of padded silence. Additionally, the convolution filters may learn to ignore the silence padded as it is deemed to not contribute to their output. This means the CNN is able to leverage the additional training data offered by the longer utterances without being hindered by the presence of padded silence.

Model	Accuracy %	F1 Score	Jaccard Score	MCC
GRU	**97.11**	0.9710	0.9440	0.9615
BiLSTM	82.99	0.8179	0.7413	0.7780
LSTM	95.32	0.9538	0.9208	0.9420
RNN	31.92	0.2141	0.1352	0.1173
CNN	96.21	0.9431	0.9125	0.9482

Table 2: *Performance on TORGO cut to timeframe of 375*

Model	Accuracy %	F1 Score	Jaccard Score	MCC
GRU	**97.18**	0.9721	0.9487	0.9646
BiLSTM	94.61	0.9455	0.8980	0.9280
LSTM	96.00	0.9595	0.9230	0.9464
RNN	58.96	0.5837	0.4191	0.4526
CNN	89.11	0.8897	0.8034	0.8555

Table 3: *Performance on TORGO cut to a timeframe of 125 (2.5 seconds so no embedded silence)*

4.2. MSDM Results

Table 4 shows the performance of the models trained on the MSDM dataset. Just as with TORGO, the best-performing model was the GRU, with it reaching an accuracy of 97.21% and F1 Score of 0.9689. This is consistent both with previous literature [5] and the results achieved on the TORGO dataset. Next, the LSTM just marginally outperformed the BiLSTM with each reaching an accuracy of 96.71% and 96.66% respectively. This indicates that processing the embeddings in both directions does not improve performance in the field of dysarthria severity assessment. Finally, the worst-performing models were the RNN achieving 68.05 % accuracy and CNN achieving 80.27 %. As with TORGO, we see that the traditional RNN had the worst performance with a Jaccard score of 0.4951 and F1 Score of 0.6437 indicating that the model also had imbalanced performance across the classes.

Relative to the results produced when training on TORGO, all RNN variants had improved performance with this difference being most evident in the traditional RNN. Again, this can be attributed to the vanishing or exploding gradient problem since MSDM has much shorter utterances, resulting in less severe gradient issues during backpropagation. In contrast to the RNN variants, the CNN's performance is significantly worse when trained on MSDM compared to TORGO. This could be attributed to their differing architectures ie. it is possible that the TORGO CNN architecture was simply better than that used for MSDM. However, these results are also consistent with the findings in Table 2 and Table 3 which indicated that the TORGO CNN was able to perform better when trained on longer utterances. Additionally, Joshy et al. found that their CNN performed significantly better when trained on the longer utterances of TORGO compared to UASpeech [5]. With MSDM utterances being considerably shorter than TORGO, it is rational that a CNN is truly more suitable when trained on TORGO than MSDM.

Model	Accuracy %	F1 Score	Jaccard Score	MCC
GRU	**97.21**	0.9689	0.9431	0.9612
BiLSTM	96.66	0.9629	0.9288	0.9509
LSTM	96.71	0.9644	0.9317	0.9525
RNN	68.05	0.6437	0.4951	0.5326
CNN	80.27	0.7978	0.6695	0.7273

Table 4: *MSDM performance*

4.3. Fine-tuned results

As discussed in Section 2.3, all models were also trained on encodings produced by a fine-tuned version of Whisper which was adapted for dysarthric speech recognition. Table 5 shows the performance of these models with the embeddings cut such that there was no silence embedded as in Table 3. We see that the performance of all models improved when trained on these fine-tuned embeddings. The most significant improvement can be seen in the traditional RNN, with its accuracy improving by over 2% and all other models improving by between 0.29% and 0.73%. Figures 4 and Figure 5 provide an explanation for these results, with the former presenting a 2D projection of the fine-tuned Whisper features while the latter shows the same for the original Whisper model features. Naturally, the plots look very similar, as the fine-tuned model was produced by only slightly altering the original model. However, we do see that the fine-tuned Whisper features can more clearly distinguish between higher dysarthria severities, meaning models trained on these features will marginally outperform those trained on normal Whisper features.

Model	Accuracy %	F1 Score	Jaccard Score	MCC
GRU	**97.47**	0.9744	0.9503	0.9661
BiLSTM	94.91	0.9489	0.9042	0.9323
LSTM	96.32	0.9629	0.9288	0.9508
RNN	61.56	0.6122	0.4213	0.4741
CNN	89.84	0.8921	0.8147	0.8693

Table 5: *Fine-tuned Whisper Results*

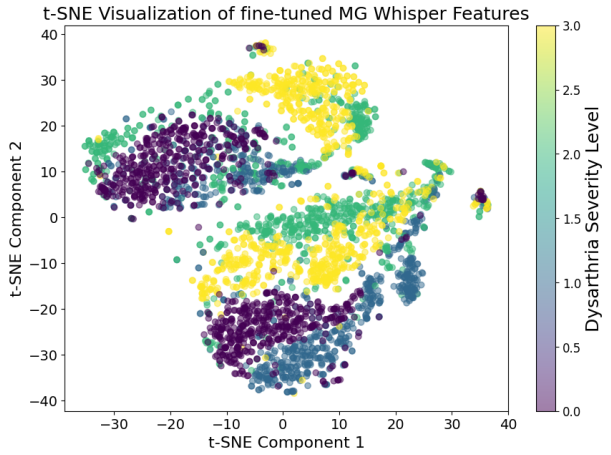

Figure 4: *2D projection of t-SNE dimensions for fine-tuned Whisper features*

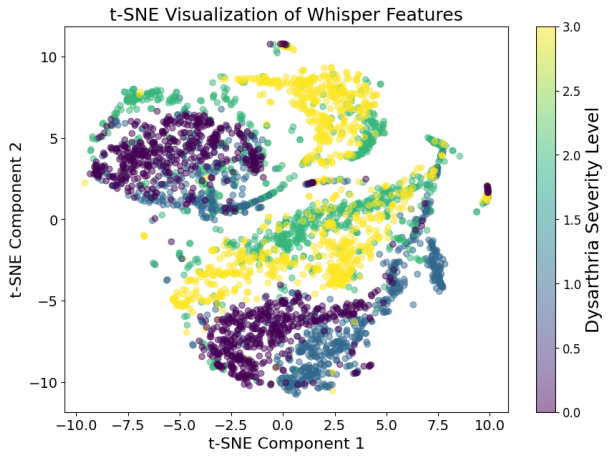

Figure 5: *2D projection of t-SNE dimensions for original Whisper features*

Table 6 shows the cumulative confusion matrix for the GRU trained on fine-tuned Whisper features. As discussed in Section 3.2, repeated testing was applied where each model was run 10 times on different train, test splits. Hence, the confusion matrix presented shows all of the predictions on the test sets across all runs. With the model achieving an average accuracy of 97.48%, the matrix unsurprisingly shows that the predicted dysarthria severity level is heavily correlated with the actual dysarthria severity level. Additionally, we see that the incorrect predictions are distributed according to their distance to the actual class. For example, 11 utterances that were labeled as having a low dysarthria severity level were predicted to be Very Low, while only 4 were predicted to have typical speech. This pattern holds across all severity levels and indicates that the model has learned the ordinal nature of the classes.

Actual \Predicted	**Typical**	**Very Low**	**Low**	**Medium**
Typical	939	7	4	0
Very Low	13	839	11	4
Low	4	11	894	14
Medium	5	9	11	907

Table 6: *Cumulative Confusion Matrix for GRU trained on fine-tuned Whisper embeddings*

5. Discussion

5.1. Comparision of results with previous literature

When comparing the results from Table 5 with dysarthria severity assessment models trained on MFCCs and CQCCs we see significant improvements in accuracy. The best-performing model presented in [5] achieved an accuracy of 96.18%, meaning the GRU trained on fine-tuned Whisper embeddings achieved an accuracy improvement of 1.29%. We also see that the LSTM and GRU achieved accuracy improvements of over 10% compared to their counterparts presented in [5] indicating that models trained on Whisper features perform better than those trained on traditional spectral features. Figure 6 illustrates this by presenting a t-SNE projection of MFCC features for the TORGO dataset. When comparing this with Figure 5 we see that the Whisper features more clearly distinguish between different dysarthria severity levels.

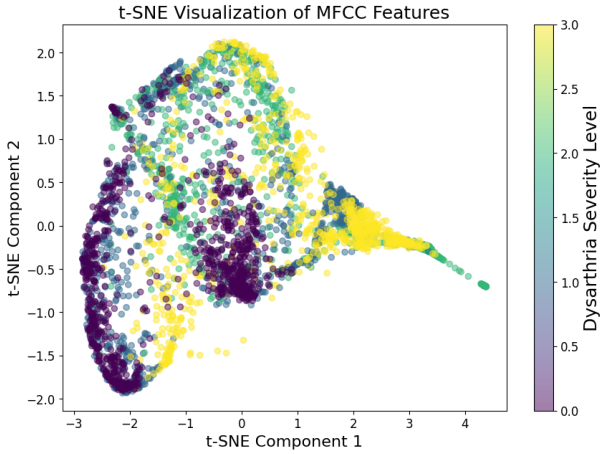

Figure 6: *2D projection of t-SNE dimensions for MFCC features*

As discussed in Section 1.2, Rathod et al. trained a CNN on Whisper embeddings and reached an accuracy of 98.49% on TORGO [10]. At first glance, this appears considerably better than the results presented in Table 5. However, the authors achieved this result by training their model 5 times on different test-train split and selecting the best result. Contrastingly, the results in this paper were achieved by averaging the performance across 10 test-train splits. The best run of the GRU trained on fine-tuned Whisper embeddings achieved an accuracy of 98.36% meaning it performed similarly to the CNN presented in [10].

5.2. Limitations

There are many potential improvements, namely in the processing of the datasets, the splitting of test and training data, the duration of training for the models, the tuning of some hyperparameters and the difference in CNN architectures across datasets. Firstly, as mentioned in Section 3.1.1, only a subset of the TORGO dataset was used. This was both to limit training time and to standardize the sequence length for the RNNs by ensuring that all utterances were at least 2.5 seconds long. However, by choosing 2.5 seconds as the cutoff point, only a relatively small subset of TORGO was used for training and the results from Table 2 and Table 3 indicate that shorter embeddings can lead to promising results. Hence, future work could explore the use of training models on a larger subset of TORGO or even the entire dataset.

Next, due to limitations in computational resources, the models were trained for an average of 32 epochs. However, previous research into dysarthria severity assessment using Whisper encodings trained for 100 epochs [8]. This effect is especially clear with training CNNs which are consistently more resource-heavy than other neural networks [15]. Another possible improvement could be made in the split between test, validation and training sets. At the moment, the split between test, train and validation is done randomly, leading to the risk that the models are overfitting to the speaking patterns of the participants. To combat this, the test and training split could have been made by using one speaker per severity class as a test set. When training the models on the TORGO dataset this would have been infeasible since there are very few speakers per severity class. However, the MSDM data set has 25 participants with dysarthria, meaning leaving one or more speakers out per class as the test set could lead to more robust models.

Additionally, improvements could be made by fine-tuning the hyperparameters of the batch size and model depth. Batch size refers to the number of training examples used to update the weights of the model per iteration. At the moment, the batch size was simply set to 4 for TORGO and 64 for MSDM using trial and error. The depth, number of hidden layers and size of hidden layers for the RNNs were also achieved in the same manner. An improvement would be to fine-tune these hyperparameters using cross-fold validation. Finally, as mentioned in Section 2.4.1, the architecture of the CNN had to be altered for the MSDM dataset as the input data had significantly different dimensions compared to TORGO due to the utterances being shorter. This means that any direct comparison of the results of the CNN across datasets must take into account this difference in model architecture.

6. Conclusion

This research aimed to answer the following research questions: "How do different ML techniques perform in dysarthria assessment using Whisper embeddings?"; "How does the inclusion of padded silence in the Whisper embeddings affect the performance of the classifiers?"; "How does training classifiers on different dysarthria datasets impact their performance?"; "How does fine-tuning Whisper to perform dysarthric ASR affect the performance of classifiers trained on its encodings?"

As discussed in Section 5, the model that achieved the greatest performance was the GRU with it reaching an accuracy of 97.11% on TORGO. Furthermore, the performance of all RNN variants improved when the Whisper embeddings were processed to not include any embedded silence with the GRU reaching 97.18% accuracy. The performance of RNN variants was improved further when trained on the MSDM dataset compared to TORGO with the GRU reaching 97.21% accuracy. In contrast, the CNN achieved its greatest accuracy of 96.21% after having it trained on the unprocessed Whisper embeddings of TORGO. Finally, the performance of all models improved when trained on fine-tuned Whisper embeddings with the GRU reaching 97.47% accuracy on TORGO.

Future work could include comparing the performance of models trained on Whisper embeddings to those trained on either unchanged or fine-tuned Wav2Vec2 embeddings. Previous studies have shown that models trained on Wav2Vec2 embeddings achieve greater levels of accuracy in dysarthria detection compared to those trained on MFCCs [22]. With all models improving performance when trained on fine-tuned Whisper embeddings compared to normal Whisper embeddings, it is plausible that this may also hold for models trained on Wav2Vec2 embeddings. Additional future work could include evaluating which dataset produced more robust models. An example of this could be training the models on one dataset and evaluating them on another, for example, training on MSDM and evaluating on TORGO. Next, as discussed in Section 4.2, the CNN had worse performance on MSDM compared to TORGO. One possible explanation would be that longer utterances include more contextual information which aids some models in dysarthria assessment. To verify this, more datasets with varying utterance lengths, such as UASpeech [7], would need to be used to train additional models and compare the results. Finally, ensemble methods such as voting or stacking could be used to combine multiple models into one, leveraging their differing architectures to produce a more accurate or robust model. Stacking involves training multiple distinct models on the same data and then training another model to integrate their outputs into a single prediction [23]. Contrastingly, voting is the process of combining the outputs of the models using some predefined rule, for example, Bayesian voting [24].

7. Responsible Research

To ensure the reproducibility of the results, repeated testing was applied with the evaluation metrics presented being averages across the train-test folds. This ensured that the results presented were truly representative of the model's performance rather than results from a favorable train-test split. Additionally, the subset of TORGO selected is clearly stated with it being all recordings from dysarthric speakers that last longer than 2.5 seconds. Finally, to ensure true reproducibility, the code for the models was made publicly available on GitHub [25].

Due to queue time limitations with Delft Blue, some results were produced by utilizing compute power from Google Colab. However, since MSDM is not a publicly available dataset, explicit permission was received from the owners of the dataset to upload it to Google Drive. To ensure additional security, only the Whisper embeddings of MSDM were uploaded, rather than the original dataset recordings. The TORGO dataset is openly available both on their website [26] and on Kaggle [27] meaning these precautions were not necessary.

8. References

[1] C. C. m. professional, "Dysarthria (slurred speech): Symptoms, causes treatment." [Online]. Available: https://my.clevelandclinic.org/health/diseases/17653-dysarthria

[2] G. Moya-Galé and E. S. Levy, "¡p¿parkinsonrsquo;s disease-associated dysarthria: prevalence, impact and management strategies¡/p¿," *Research and Reviews in Parkinsonism*, vol. 9, p. 9–16, May 2019.

[3] P. Enderby, *Chapter 22 - Disorders of communication: dysarthria*, ser. Neurological Rehabilitation. Elsevier, Jan. 2013, vol. 110, p. 273–281. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444529015000228

[4] ——, "Frenchay dysarthria assessment," *British Journal of Disorders of Communication*, Jan. 1980. [Online]. Available: https://www.tandfonline.com/doi/abs/10.3109/13682828009112541

[5] A. A. Joshy and R. Rajan, "Automated dysarthria severity classification: A study on acoustic features and deep learning techniques," *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 30, p. 1147–1157, 2022.

[6] F. Rudzicz, A. Namasivayam, and T. Wolff, "The torgo database of acoustic and articulatory speech from speakers with dysarthria," *Language Resources and Evaluation*, vol. 46, p. 1–19, Jan. 2010.

[7] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, K. Watkin, and S. Frame, "Dysarthric speech database for universal access research," Sep. 2008, p. 1741–1744.

[8] S. Rathod, M. Charola, and H. A. Patil, "Noise robust whisper features for dysarthric severity-level classification," in *Pattern Recognition and Machine Intelligence*, P. Maji, T. Huang, N. R. Pal, S. Chaudhury, and R. K. De, Eds. Cham: Springer Nature Switzerland, 2023, p. 708–715.

[9] S. R. Mani Sekhar, G. Kashyap, A. Bhansali, A. A. A., and K. Singh, "Dysarthric-speech detection using transfer learning with convolutional neural networks," *ICT Express*, vol. 8, no. 1, p. 61–64, Mar. 2022.

[10] S. Rathod, M. Charola, A. Vora, Y. Jogi, and H. A. Patil, "Whisper features for dysarthric severity-level classification," 2023, p. 1523–1527. [Online]. Available: https://www.isca-archive.org/interspeech_2023/rathod23_interspeech.html

[11] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision."

[12] J. Liu, "Audio-video database from subacute stroke patients for dysarthric speech intelligence assessment and preliminary analysis," *Biomedical Signal Processing and Control*, 2023.

[13] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Feb. 2008, vol. 2.

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," no. arXiv:2106.09685, Oct. 2021, arXiv:2106.09685 [cs]. [Online]. Available: http://arxiv.org/abs/2106.09685

[15] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," no. arXiv:1511.08458, Dec. 2015, arXiv:1511.08458 [cs]. [Online]. Available: http://arxiv.org/abs/1511.08458

[16] [Online]. Available: https://www.upgrad.com/blog/basic-cnn-architecture/

[17] R. M. Schmidt, "Recurrent neural networks (rnns): A gentle introduction and overview," no. arXiv:1912.05911, Nov. 2019, arXiv:1912.05911 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1912.05911

[18] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, p. 2222–2232, Oct. 2017, arXiv:1503.04069 [cs].

[19] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, p. 2673–2681, Nov. 1997.

[20] R. Cahuantzi, X. Chen, and S. Güttel, "A comparison of lstm and gru networks for learning symbolic sequences," in *Intelligent Computing*, K. Arai, Ed. Cham: Springer Nature Switzerland, 2023, p. 771–785.

[21] K. T. Mengistu and F. Rudzicz, "Adapting acoustic and lexical models to dysarthric speech," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Prague, Czech Republic: IEEE, May 2011, p. 4924–4927. [Online]. Available: http://ieeexplore.ieee.org/document/5947460/

[22] F. Javanmardi, S. Tirronen, M. Kodali, S. R. Kadiri, and P. Alku, "Wav2vec-based detection and severity level classification of dysarthria from speech," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, p. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10094857?casa_token=70d4IbMfcNAAAAAA:cHMKLeRtrnfPW_okNU8AS5HFbn0g7yEhRXuW_MclhpAzE2MV1IO7c-c7sDz8BKsDAa5iCZMR

[23] R. Dey and R. Mathur, "Ensemble learning method using stacking with base learner, a comparison," in *Proceedings of International Conference on Data Analytics and Insights, ICDAI 2023*, N. Chaki, N. D. Roy, P. Debnath, and K. Saeed, Eds. Singapore: Springer Nature, 2023, p. 159–169.

[24] T. G. Dietterich, "Ensemble methods in machine learning," *Multiple Classifier Systems*, p. 1–15, 2000.

[25] C. Charlesworth, "Dysarthria severity classifiers on Whisper embeddings repository," https://github.com/ChrisCharlesworth/WhisperBasedClassifiers, Jun. 2023.

[26] [Online]. Available: https://www.cs.toronto.edu/~complingweb/data/TORGO/torgo.html

[27] [Online]. Available: https://www.kaggle.com/datasets/pranaykoppula/torgo-audio