Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

A Self-Training Approach for Whisper to Enhance Long Dysarthric Speech Recognition

*Shiyao Wang, Jiaming Zhou, Shiwan Zhao, Yong Qin**

College of Computer Science, Nankai University, China

wangshiyao@mail.nankai.edu.cn

Abstract

Dysarthric speech recognition (DSR) enhances the accessibility of smart devices for dysarthric speakers with limited mobility. Previously, DSR research was constrained by the fact that existing datasets typically consisted of isolated words, command phrases, and a limited number of sentences spoken by a few individuals. This constrained research to command-interaction systems and speaker adaptation. The Speech Accessibility Project (SAP) changed this by releasing a large and diverse English dysarthric dataset, leading to the SAP Challenge to build speaker- and text-independent DSR systems. We enhanced the Whisper model's performance on long dysarthric speech via a novel self-training method. This method increased training data and adapted the model to handle potentially incomplete speech segments encountered during inference. Our system achieved second place in both Word Error Rate and Semantic Score in the SAP Challenge.

Index Terms: dysarthric speech recognition, Whisper, self-training, long speech, segmentation

1. Introduction

Dysarthria, a speech disorder stemming from neurological conditions like Parkinson's disease and cerebral palsy, affects vocal control and leads to speech impediments such as stuttering, abnormal pauses, altered prosody, and mispronunciation. These characteristics pose significant challenges for automatic speech recognition (ASR) systems, which are primarily trained on typical speech [1]. Although modern ASR systems [2, 3] demonstrate robust performance on typical speech due to abundant training data, dysarthric speech recognition (DSR) suffers from data scarcity. This limitation restricts DSR research to word-level [4] and command-word level [5] tasks. Widely used English dysarthric datasets like UASpeech [4] and Torgo [6] lack the speaker and text diversity necessary for building robust speaker-independent (SI) and text-independent (TI) DSR systems. Consequently, previous DSR research focuses on data augmentation [7, 8] and developing personalized models [9, 10, 11, 12]. To address the data scarcity issue and stimulate more advanced research, Hasegawa-Johnson et al. [13] launched the Speech Accessibility Project (SAP) to collect a large and diverse dataset of dysarthric English speech and subsequently organized the SAP Challenge.

As participants in the SAP Challenge, we developed SI and TI DSR models using the SAP 2024-04-30 (SAP0430_processed) dataset[1]. Building on the findings of Zheng et al. [14] and Singh et al. [15], who demonstrated the effectiveness of fine-tuning models like wav2vec 2.0 [16] and Whisper [2] on previous SAP data (with Whisper showing superior performance), we prioritized base model selection. We compared Whisper against other advanced ASR technologies not yet evaluated on the SAP data, including HuBERT [17], and pre-trained models from ESPnet [18] and Wenet [19]. Due to Challenge time constraints and GPU utilization efficiency, we trained on short speech segments (see Section 3.2 for specific training speech duration limits). Furthermore, because Whisper's audio encoder truncates long spectrograms to fit its positional encoding (leading to incomplete predictions)[20], we also evaluated model performance with an even segmentation (E-S) strategy during inference (detailed in Section 3.2.2). Ultimately, the fine-tuned Whisper large-v3, with the E-S strategy employed for inference, demonstrated superior performance (Table 1). While Whisper [2] uses a buffering approach for long-speech transcription, segmenting audio into 30-second chunks and using predicted timestamps to advance the window, WhisperX [21] has shown that Whisper's timestamp predictions are often inaccurate, limiting long-speech recognition. WhisperX combines timestamp-free decoding with Voice Activity Detection (VAD) for improved results. Specifically, WhisperX pre-segments audio using VAD, concatenates VAD segments into approximately 30-second blocks, and then performs transcription. Inspired by WhisperX, we explored various inference strategies based on timestamp-free decoding (Table 2).

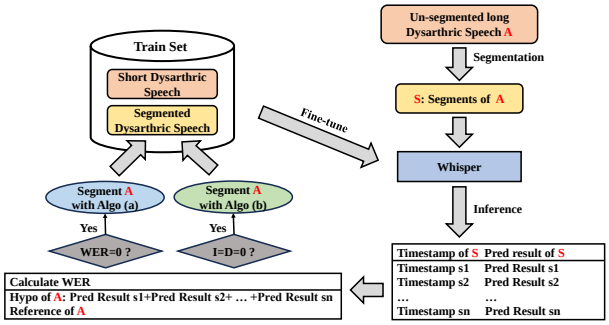

Figure 1: *The overview of our self-training approach to segment long dysarthric speech (ST-SLDS). 'Hyo': 'hypothesis'. Algo (a) and Algo (b) see Fig 2.*

While various strategies can improve long-speech inference, Whisper's architecture limits fine-tuning to data segments of 30 seconds or less. This constraint introduces several drawbacks: (1) Long-speech data loss: Manual segmentation and transcript matching are possible [22], but the labor cost is excessive. (2) Training-inference mismatch: Segmenting long speech during inference can lead to incomplete sentences in the input audio. As Niehues et al. [23] demonstrated in machine translation, models trained only on complete data tend to generate

* Corresponding author.

[1] Sample durations range from 0.039 seconds to 121.02 seconds.

10.21437/Interspeech.2025-934

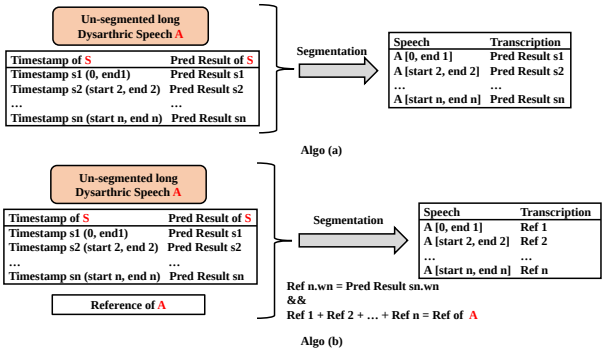

Figure 2: *Segmentation algorithm (a) and (b). 'wn' represents the number of words. 'Ref': 'Reference'. 'Algo': 'Algorithm'.*

complete sentences even from partial inputs. Liu et al. [24] observed similar mismatches in speech recognition. Both studies addressed this by incorporating "partial sequence" data into training. Following their approach, we attempted to incorporate segmented data to mitigate the training-inference inconsistency. We segmented the unused long dysarthric speech and annotated it using a self-training method.

Self-training, a common semi-supervised learning method [25, 26, 27, 28, 29, 30], is employed when labeled data is less and unlabeled speech data is abundant. It aims to transform unlabeled data into usable training material, thereby enhancing the model's generalization, robustness, and adaptability. Self-training operates within a teacher-student framework. This framework involves training a teacher model on labeled data, generating pseudo-labels for unlabeled data, and then filtering the unlabeled data based on the confidence of the pseudo-labels [31] (e.g., frame-level, utterance-level) or through heuristics [25]. The filtered data is then combined with the labeled data to train a student model, which subsequently becomes the new teacher, and the process is repeated iteratively. In the context of improving Whisper's performance on long dysarthric speech, we applied this self-training concept. Specifically, we designed an iterative training process within a teacher-student framework (Figure 1). First, we built a teacher model using the short duration speech sample data. Then, we carried out segmentation-based inference on the long audio to generate pseudo-labels. Subsequently, we screened the data based on a heuristic strategy and refined the annotations using designed segmentation algorithms (Figure 2). Finally, we added the segmented data to the training set for a new iteration.

The main **contributions** of this work are as follows:

- We tested different advanced ASR technologies and various inference settings to build a speaker-independent and text-independent DSR model with strong generalization ability.
- We propose a self-training approach specifically designed to enhance Whisper's ability to process long dysarthric speech. This method simultaneously increases the available training data and mitigates the training-inference mismatch, leading to improved performance.

2. Whisper self-training optimization

We propose a self-training approach to segment long dysarthric speech (ST-SLDS, see Figure 1). First, we fine-tune the Whisper model on short dysarthric speech data to create a teacher model. In subsequent iterations, the fine-tuning dataset includes segmented data from previous iterations. The teacher model then performs inference on long dysarthric speech. Prior to inference, the long speech is segmented using a segmentation strategy to obtain timestamped segments, which are then individually fed into the teacher model to generate corresponding predictions. To select high-quality data, we use a heuristic screening strategy: (1) We reconstruct a prediction for the original long speech by combining the predictions of its segments, and then calculate its WER against the reference transcription. (2) We select data with either a WER of 0 (WER = 0) or a non-zero WER but zero insertion and deletion errors (I = D = 0). Data with a non-zero WER is re-tested in later iterations. The selected data is then segmented and annotated using the algorithms detailed in Figure 2. Data filtered by the "WER = 0" condition uses Algorithm (a), segmenting the speech at the predicted timestamps and using the corresponding prediction as the label. Data filtered by the "I = D = 0" condition uses Algorithm (b), also segmenting at the predicted timestamps but generating labels by extracting the same number of words from the reference transcription, matching the segment's position in the original speech. Finally, the segmented data is added to the training set, and the process is repeated iteratively.

Table 1: *Performance comparison in WER (%) among different ASR models. 'E-S': even segmentation. 'W-' models are different size variants of the Whisper model.*

Model Setting	HuBERT CTC	ESPnet Pretrain 1	ESPnet Pretrain 2	Wenet Pretrain
Params	94.5 M	148.9 M	148.9 M	131.3 M
FT	17.7800	10.6203	**8.8852**	8.8986
FT + E-S	18.0746	25.3420	24.2847	8.8700
	W-Small	**W-Medium**	**W-Large-v2**	**W-Large-v3**
Params	241.7 M	763.6 M	1543.3 M	1543.5 M
FT	16.9325	14.3460	13.4699	13.2869
FT + E-S	12.3040	9.7766	8.0043	**7.6106**

3. Experiments

3.1. Dataset

The SAP Challenge data comprises the following components: (1) SAP0430_processed: This dataset has been preprocessed through 16kHz audio resampling and text regularization and is divided into training (290.4 hours, 131,420 samples, 369 speakers) and development (dev) (43.6 hours, 19,275 samples, 55 speakers) partitions. The dev partition was further subdivided by us into a validation set (9,275 samples, 55 speakers) and the 0430Test set (10,000 samples, 54 speakers). Our models are trained using disfluency-removed transcripts. (2) SAP-1130 (released on November 30, 2024): This dataset consists of unprocessed 48kHz audio and unregulated text. We preprocessed the data newly added in the SAP-1130 training partition compared to the SAP0430_processed training partition. This preprocessed data was then used as the 1130Test set (11 hours, 2,160 samples, 27 speakers). (3) The SAP Challenge evaluates submissions on Test 1 (7,601 samples) and Test 2 (8,043 samples). These evaluation sets are inaccessible to us and exhibit no speaker or text overlap with the data we obtained.

3.2. Experimental settings

3.2.1. Model and training settings

HuBERT CTC: We utilized the pre-trained HuBERT Base model[2] and incorporated a CTC layer to build DSR model.

[2] https://dl.fbaipublicfiles.com/hubert/hubert_base_ls960.pt

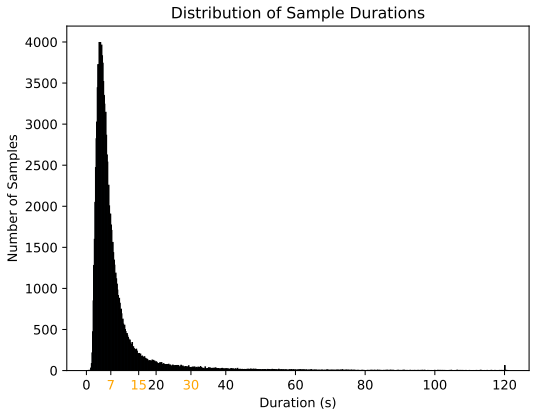

Figure 3: *The duration distribution of the training data in SAP0430_processed.*

Training was conducted under the Fairseq framework[3], based on the hyperparameters outlined in the base_10h.yaml configuration. We set the learning rate to 10^{-5} and the warmup steps to 32,000, halting the training when the validation set loss did not decrease over 10 consecutive epochs.

ESPnet Pretrain: We fine-tuned two separate models within the ESPnet framework. The first model was pre-trained on the 960-hour LibriSpeech dataset[4] (ESPnet Pretrain 1), while the second was pre-trained on the 10,000-hour GigaSpeech dataset[5] (ESPnet Pretrain 2). Both models were fine-tuned for 100 epochs using the train_asr_e_branchformer.yaml configuration.

Wenet Pretrain: We fine-tuned a model provided by Wenet, which was pre-trained on 10,000-hour GigaSpeech data[6], for 100 epochs using the pre-trained configuration file.

Whisper By leveraging the Whisper-Flamingo codebase[7], we fine-tuned several Whisper pre-trained models (small, medium, large-v2, large-v3). All models were fine-tuned using hyperparameters specified in the corresponding configuration files (see config/audio/ in the repository). Given that Whisper [2] was pre-trained on original collected text, during which machine-transcribed data (all-upper or all-lower) was filtered out. To align with Whisper's pre-training data characteristics, we evaluated two transcription formats during training: (1) A-U: all uppercase, a common format for ASR model fine-tuning; (2) F-U: the first letter in uppercase and the rest in lowercase.

Training speech duration limits Regarding models with fewer parameters compared to Whisper, we limited the maximum speech length to 30 seconds and maximized the batch size on an A100 GPU. The large parameter count of the Whisper large-v3 model restricted the maximum batch size to 2 under the 30-second duration limit [32], leading to slow training progress and insufficient GPU utilization. By analyzing the duration distribution of the training data in SAP0430_processed dataset (Figure 3), we set a 15-second duration limit for Whisper, which is twice the duration of the highest data concentration.

3.2.2. Inference strategies

Segmentation strategies:

[3] https://github.com/facebookresearch/fairseq/blob/main/examples/hubert/README.md

[4] https://huggingface.co/asapp/e_branchformer_librispeech

[5] https://huggingface.co/pyf98/gigaspeech_e_branchformer

[6] https://github.com/wenet-e2e/wenet/blob/main/docs/pretrained_models.md

[7] https://github.com/roudimit/whisper-flamingo

Table 2: *Performance comparison in WER (%) and SemScore among different inference settings. A detailed description of the inference settings can be found in Section 3.2.*

Txn		Segmentation		Decoding			Result	
A-U	F-U	E-S	VAD-S	G-S	B-S	Prompt	WER	SemScore
✓		✓		✓			7.6106	92.3187
✓		✓		✓		1	7.8871	92.3996
✓		✓			5		7.4095	92.3653
✓		✓			10		**7.3694**	92.4093
✓		✓			10	1	7.4504	92.4863
✓		✓			10	2	7.5096	**92.4873**
✓		✓			20		7.3932	92.4088
✓		✓			30		**7.3685**	92.4324
	✓	✓			10		6.8422	93.2347
	✓		✓		10		**6.3245**	93.5297
	✓		✓		10	1	6.3884	**93.5883**

- **Even Segmentation (E-S)**: Long speech is evenly partitioned into segments. The number of segments (n) is determined by:

$$n = N//(L_{max} \times 16000) + 1, \quad (1)$$

  where N is the number of audio sampling points and L_{max} is the maximum input duration.
- **VAD Segmentation (VAD-S)**: We utilize Silero VAD[8] to generate the timestamps of speech segments. In contrast to WhisperX, we preserve all segments, even those without detected speech. Segmentation begins at timestamp 0, with subsequent points determined by the VAD's detected speech start timestamps. An iterative process ensures the time interval between segmentation points remains close to, but does not exceed L_{max}. Given that Silero VAD is not specifically trained for dysarthric speech, its detection may be inaccurate, and in some cases, it may fail to detect any speech segments. If this occurs, we employ the E-S strategy for segmentation.

Decoding strategies:

- Greedy Search (G-S)
- Beam Search (B-S): Beam size is a configurable parameter (e.g., 5, 10, 20, 30).
- Prompt: For segmented speech, the result of the previous segment is used as a prompt for Whisper's inference of the current segment.

3.2.3. Evaluation metrics

The SAP Challenge uses two evaluation metrics: Word Error Rate (WER) and Semantic Score (SemScore). WER quantifies the normalized edit distance between the reference transcript and the DSR system's output. SemScore combines BERTScore [33], phonetic distance, and natural language inference probability into a single metric. A lower WER and a higher SemScore indicate superior performance. See the Challenge's template repository[9] for detailed metric calculations.

3.3. Results and discussions

Table 1 compares the performance of DSR models fine-tuned from different pre-trained models on 0430Test. In direct inference (without segmentation), the fine-tuned ESPnet Pretrain 2 model achieved the best results. However, Whisper's performance was limited by its encoder's truncation of long spectro-

[8] https://github.com/snakers4/silero-vad

[9] https://github.com/xiuwenz2/SAPC-template

Table 3: *The effect of enhancing Whisper by using our ST-SLDS approach. '1st'-'4th' denote the 1st-4th iterations of the self-training method. 'Add dev' means incorporating the dev part of SAP0430_processed into the training set.*

Settings	**Train**	**0430Test**		**1130Test**		**Test 1 Result**		**Test 2 Result**	
	Samples	**WER**	**SemScore**	**WER**	**SemScore**	**WER**	**SemScore**	**WER**	**SemScore**
Baseline	119845	6.8422	93.2347	8.8376	89.1004	9.2663	88.3354	11.1954	84.9895
1st WER=0	124556	5.8011	93.7636	7.9411	89.9006	—	—	—	—
1st WER=0 + I=D=0	128094	5.7878	93.8621	7.8882	90.0194	—	—	—	—
2nd WER=0	128974	5.6362	93.9558	7.6172	90.3785	—	—	—	—
2nd WER=0 + I=D=0	133988	5.5628	**94.1113**	7.5798	90.2510	9.1918	88.3758	11.1487	85.0233
3rd WER=0	130624	5.6162	93.8716	7.6150	90.2384	—	—	—	—
3rd WER=0 + I=D=0	136325	**5.5533**	93.9591	**7.4675**	**90.4589**	8.7301	89.3272	10.3923	85.9151
4th WER=0	131339	5.6162	93.9340	7.6040	90.3504	—	—	—	—
4th WER=0 + I=D=0	137415	5.5571	93.9712	7.5754	90.2702	—	—	—	—
Add dev	137248	3.4635	97.9971	9.0667	89.1355	—	—	—	—
1st WER=0 , Add dev	141959	3.1079	98.1854	8.1283	89.7760	—	—	—	—
1st WER=0 + I=D=0 , Add dev	145496	3.0088	98.1669	7.7758	89.9184	—	—	—	—
2nd WER=0 , Add dev	147444	2.9392	98.1726	7.6877	90.3077	—	—	—	—
2nd WER=0 + I=D=0 , Add dev	153509	2.6170	98.4272	7.6415	**90.3769**	8.4151	89.6910	10.3091	86.0170
3rd WER=0 , Add dev	150970	2.8820	98.2746	7.6327	90.1662	—	—	—	—
3rd WER=0 + I=D=0 , Add dev	158246	2.5559	98.4155	**7.5820**	90.0180	**8.2479**	**89.7438**	**10.0294**	**86.1884**
4th WER=0 , Add dev	152358	2.5502	**98.4969**	7.8001	89.7829	8.5954	89.2178	10.4831	85.6587
4th WER=0 + I=D=0 , Add dev	160072	**2.5493**	98.4856	7.6459	90.2306	8.5507	89.1929	10.3536	85.7684

grams, resulting in incomplete predictions. To accurately assess Whisper's potential, we implemented a simple E-S strategy during inference. This significantly improved Whisper's performance ($p < 0.05$), with the fine-tuned Whisper large-v3 model achieving the best results using the E-S strategy. Conversely, applying the E-S strategy degraded the performance of other models, except for Wenet Pretrain. Finally, we selected fine-tuned Whisper large-v3 as our base model and adopted the segmentation strategy for long speech during inference.

Table 2 presents a comparison of the performance of different inference settings on 0430Test. To isolate the impact of decoding strategies, we first tested them using the A-U fine-tuned model and the simple E-S strategy. Beam search outperformed greedy search, with a beam size of 10 proving optimal. Although increasing the beam size to 30 slightly decreased the WER, local testing indicated that it exceeded the Challenge's 240-minute limit. Implementing the Prompt strategy with a beam size of 10 increased the SemScore but worsened the WER. Connecting the predictions of the previous two segments as a prompt (Prompt = 2) further increased the SemScore slightly, yet the negative impact on the WER was so substantial that we only considered Prompt = 1. Subsequently, we tested improved inference strategies on the model fine-tuned with the F-U setting, which yielded significantly better performance than the A-U fine-tuned model, indicating the advantage of aligning the fine-tuning setting with the pre-training setting. Replacing the E-S strategy with the VAD-S strategy, which provides better segmentation points, further enhanced performance. We also tested inferring only speech segments, similar to WhisperX (The results were not presented in Table 2). This approach produced performance inferior to that of the VAD-S strategy. Presumably, since the VAD tool was not adapted to dysarthria, it failed to accurately detect speech segments in dysarthric speech. Therefore, we chose to fine-tune the model based on the F-U setting, set the beam size to 10 during inference, and use the VAD-S strategy. We didn't adopt the Prompt strategy as it had slow inference and worsened WER.

We further improved performance by augmenting the training data using our ST-SLDS method to segment long dysarthric speech and incorporating the SAP0430_processed dev partition (including 0430Test). Initially, tests were conducted without the dev partition. As shown in Table 3, during the first three iterations, the WER for both 0430Test and 1130Test steadily decreased, indicating the effectiveness of the ST-SLDS method. However, in the fourth iteration, this trend changed. The remaining long dysarthric speech was highly challenging, making it difficult to meet our filtering conditions. Consequently, minimal new data was added. Furthermore, due to our method's re-filtering of "I = D = 0" data in the next iteration, duplicate data emerged in the new data pool. We hypothesize that these duplicates induced overfitting in the model during the fourth iteration, leading to a performance rebound. Therefore, we halted further iterations. Subsequently, we introduced the dev partition data into the training process. Since 0430Test was integrated into the training set, its performance is no longer a reference point. Notably, the performance of 1130Test under this new setting was lower compared to the settings without the dev partition. We attribute this to the dev partition substantially increasing the number of speakers and the overall complexity of the learning task. Given the existing speaker overlap between 1130Test and the original training set, the altered speaker distribution likely led to the decline in model performance. We submitted the models with relatively superior performance under both settings (with and without the dev partition). The results from Test 1 and Test 2 clearly demonstrate the effectiveness of both our strategies for augmenting the training data. Furthermore, submitting the model from the fourth iteration with the dev partition conclusively verified that three iterations represented the optimal choice.

4. Conclusions

To improve Whisper's recognition of long dysarthric speech, we propose a self-training method that segments and labels long speech, enabling its use for fine-tuning and mitigating training-inference mismatch. By segmenting the data, we simulate the incomplete speech segments encountered during inference. Ultimately, our constructed speaker-independent and text-independent dysarthric speech recognition system ranked second in both the WER and SemScore of the SAP Challenge.

5. Acknowledgements

This work has been supported in part by NSF China (Grant No.62271270).

6. References

[1] L. De Russis and F. Corno, "On the impact of dysarthric speech on contemporary asr cloud platforms," *Journal of Reliable Intelligent Environments*, vol. 5, pp. 163–172, 2019.

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *ICML*. PMLR, 2023, pp. 28 492–28 518.

[3] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang *et al.*, "Google usm: Scaling automatic speech recognition beyond 100 languages," *arXiv preprint arXiv:2303.01037*, 2023.

[4] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. S. Huang, K. Watkin, and S. Frame, "Dysarthric speech database for universal access research," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[5] R. Turrisi, A. Braccia, M. Emanuele, S. Giulietti, M. Pugliatti, M. Sensi, L. Fadiga, and L. Badino, "Easycall corpus: a dysarthric speech dataset," *arXiv preprint arXiv:2104.02542*, 2021.

[6] F. Rudzicz, A. K. Namasivayam, and T. Wolff, "The torgo database of acoustic and articulatory speech from speakers with dysarthria," *Language resources and evaluation*, vol. 46, 2012.

[7] M. Soleymanpour, M. T. Johnson, R. Soleymanpour, and J. Berry, "Synthesizing dysarthric speech using multi-speaker tts for dysarthric speech recognition," in *ICASSP*. IEEE, 2022.

[8] W.-Z. Leung, M. Cross, A. Ragni, and S. Goetze, "Training data augmentation for dysarthric automatic speech recognition by text-to-dysarthric-speech synthesis," in *Interspeech*, 2024.

[9] J. Shor, D. Emanuel, O. Lang, O. Tuval, M. Brenner, J. Cattiau, F. Vieira, M. McNally, T. Charbonneau, M. Nollstadt *et al.*, "Personalizing asr for dysarthric and accented speech with limited data," *arXiv preprint arXiv:1907.13511*, 2019.

[10] Z. Qian, K. Xiao, and C. Yu, "A survey of technologies for automatic dysarthric speech recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2023, no. 1, p. 48, 2023.

[11] S. Wang, S. Zhao, J. Zhou, A. Kong, and Y. Qin, "Enhancing dysarthric speech recognition for unseen speakers via prototype-based adaptation," in *Interspeech 2024*, 2024, pp. 1305–1309.

[12] S. Wang, J. Zhou, S. Zhao, and Y. Qin, "Pb-lrdwws system for the slt 2024 low-resource dysarthria wake-up word spotting challenge," in *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2024, pp. 586–591.

[13] M. Hasegawa-Johnson, X. Zheng, H. Kim, C. Mendes, M. Dickinson, E. Hege, C. Zwilling, M. M. Channell, L. Mattie, H. Hodges, L. Ramig, M. Bellard, M. Shebanek, L. Sar, K. Kalgaonkar, D. Frerichs, J. P. Bigham, L. Findlater, C. Lea, S. Herrlinger, P. Korn, S. Abou-Zahra, R. Heywood, K. Tomanek, and B. MacDonald, "Community-supported shared infrastructure in support of speech accessibility," *Journal of Speech, Language, and Hearing Research*, vol. 67, no. 11, pp. 4162–4175, 2024. [Online]. Available: https://pubs.asha.org/doi/abs/10.1044/2024_JSLHR-24-00122

[14] X. Zheng, B. Phukon, and M. Hasegawa-Johnson, "Fine-tuning automatic speech recognition for people with parkinson's: An effective strategy for enhancing speech technology accessibility," *arXiv preprint arXiv:2409.19818*, 2024.

[15] S. Singh, Q. Wang, Z. Zhong, C. Mendes, M. Hasegawa-Johnson, W. Abdulla, and S. R. Shahamiri, "Robust cross-etiology and speaker-independent dysarthric speech recognition," *arXiv preprint arXiv:2501.14994*, 2025.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[17] 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," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[18] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[19] Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, and X. Lei, "Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," *arXiv preprint arXiv:2102.01547*, 2021.

[20] J. Zhou, S. Zhao, J. He, H. Wang, W. Zeng, Y. Chen, H. Sun, A. Kong, and Y. Qin, "M2r-whisper: Multi-stage and multi-scale retrieval augmentation for enhancing whisper," *arXiv preprint arXiv:2409.11889*, 2024.

[21] M. Bain, J. Huh, T. Han, and A. Zisserman, "Whisperx: Time-accurate speech transcription of long-form audio," *arXiv preprint arXiv:2303.00747*, 2023.

[22] G. Sanguedolce, D.-C. Gruia, S. Brook, P. Naylor, and F. Geranmayeh, "Universal speech disorder recognition: Towards a foundation model for cross-pathology generalisation," in *Advancements In Medical Foundation Models: Explainability, Robustness, Security, and Beyond*.

[23] J. Niehues, N.-Q. Pham, T.-L. Ha, M. Sperber, and A. Waibel, "Low-latency neural speech translation," *arXiv preprint arXiv:1808.00491*, 2018.

[24] D. Liu, G. Spanakis, and J. Niehues, "Low-latency sequence-to-sequence speech recognition and translation by partial hypothesis selection," *arXiv preprint arXiv:2005.11185*, 2020.

[25] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," in *ICASSP*. IEEE, 2020, pp. 7084–7088.

[26] Y. Long, Y. Li, S. Wei, Q. Zhang, and C. Yang, "Large-scale semi-supervised training in deep learning acoustic model for asr," *IEEE Access*, vol. 7, pp. 133 615–133 627, 2019.

[27] S. H. K. Parthasarathi and N. Strom, "Lessons from building acoustic models with a million hours of speech," in *ICASSP*. IEEE, 2019, pp. 6670–6674.

[28] M. Bartelds, N. San, B. McDonnell, D. Jurafsky, and M. Wieling, "Making more of little data: Improving low-resource automatic speech recognition using data augmentation," *arXiv preprint arXiv:2305.10951*, 2023.

[29] G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," *arXiv preprint arXiv:1911.08460*, 2019.

[30] J. Zhou, S. Zhao, N. Jiang, G. Zhao, and Y. Qin, "Madi: Inter-domain matching and intra-domain discrimination for cross-domain speech recognition," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[31] K. Veselỳ, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 267–272.

[32] Y. Peng, J. Tian, B. Yan, D. Berrebbi, X. Chang, X. Li, J. Shi, S. Arora, W. Chen, R. Sharma *et al.*, "Reproducing whisper-style training using an open-source toolkit and publicly available data," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[33] J. Tobin, Q. Li, S. Venugopalan, K. Seaver, R. Cave, and K. Tomanek, "Assessing asr model quality on disordered speech using bertscore," *arXiv preprint arXiv:2209.10591*, 2022.