Proceedings of the RaPID-4 @LREC 2022, pages 71–79
Marseille, 25 June 2022

Data Augmentation for the Post-Stroke Speech Transcription (PSST) Challenge: Sometimes Less is More

Jiahong Yuan, Xingyu Cai, Kenneth Church
Baidu Research USA
1195 Bordeaux Dr, Sunnyvale, CA 94089, USA
{jiahongyuan, xingyucai, kennethchurch}@baidu.com

Abstract

We employ the method of fine-tuning wav2vec2.0 for recognition of phonemes in aphasic speech. Our effort focuses on data augmentation, by supplementing data from both in-domain and out-of-domain datasets for training. We found that although a modest amount of out-of-domain data may be helpful, the performance of the model degrades significantly when the amount of out-of-domain data is much larger than in-domain data. Our hypothesis is that fine-tuning wav2vec2.0 with a CTC loss not only learns bottom-up acoustic properties but also top-down constraints. Therefore, out-of-domain data augmentation is likely to degrade performance if there is a language model mismatch between "in" and "out" domains. For in-domain audio without ground truth labels, we found that it is beneficial to exclude samples with less confident pseudo labels. Our final model achieves 16.7% PER (phoneme error rate) on the validation set, without using a language model for decoding. The result represents a relative error reduction of 14% over the baseline model trained without data augmentation. Finally, we found that "canonicalized" phonemes are much easier to recognize than manually transcribed phonemes.

Keywords: wav2vec2.0, aphasia, phoneme recognition, data augmentation

1. Introduction

The diagnosis of post-stroke language disorders, namely aphasia, depends on recognizing phonemes in speech. For example, reduced activation of lexical-semantic representations in aphasia may result in producing "dog" for the target word "cat", while reduced activation of phonological representations may result in producing "dog" for the target word "log" (Foygel and Dell, 2000). The primary task of the Post-Stroke Speech Transcription (PSST) Challenge (Task A) is to develop an automatic phoneme recognition system that accurately identifies the phonemes produced by subjects with aphasia. The phonemes they actually produce may differ in important ways from the words they intended to produce. This paper describes our effort for the task.

Recognizing phonemes in aphasic speech is a challenging task for both human judges and computers. Different types of aphasia are associated with different types of linguistic symptoms (Wilson et al., 2010). Problems such as disfluencies, mispronunciations, and articulation deficits create interesting challenges for automatic phoneme recognition. In addition, limitations in data availability introduce additional challenges. State-of-the-art models tend to be more effective when there is plenty of in-domain data with ground-truth labels (with little room for inter-annotator disagreements).

This paper fine-tunes wav2vec2.0 for Task A of the PSST Challenge. For recognition of speech from healthy speakers, the wav2vec2.0 model has recently achieved impressive results. But how well does this approach transfer to speech from the PSST challenge?

Our effort focuses on data augmentation, by supplementing data from both in-domain and out-of-domain datasets for training. We found that modest amounts of out-of-domain data can improve performance, but too much of a good thing is not necessarily a good thing. In particular, performance degrades significantly when there is much more out-of-domain data than in-domain data.

Datasets vary in many respects. Some are in-domain and some are out-of-domain. Some come with better ground truth labels than others. Different annotation methods are used by different researchers. Some datasets do not provide ground truth labels.

When there are no ground truth labels, we use pseudo-labels. That is, use predictions from a trained model as if they are gold labels. Iterating the self-training process leads to improve performance, especially when utterances with low confidence are removed from the self-training process.

Less is more. That is, we found that data augmentation can be helpful, but not if there is too much out-of-domain data relative to in-domain data, or if there are too many pseudo-labels of dubious quality. Our final model achieves 16.7% PER (phoneme error rate) on the validation set, without using a language model. The result represents a relative error reduction of 14% over the baseline model trained without data augmentation.

2. Previous Work

2.1. Finetuning wav2vec2.0 for ASR

Wav2vec2.0 (Baevski et al., 2020) is a Transformer-based framework for self-supervised learning of speech representations from raw audio data. The speech signal is processed by a multilayer convolutional network to obtain latent features at every 25 ms, which are

then fed into vector quantization and Transformer networks. The contextualized representations from pretrained wav2vec2.0 capture a rich amount of information about speech, demonstrated by probing experiments showing that the representations can perform well on a wide range of tasks (Ma et al., 2021; Shah et al., 2021).

Pre-trained wav2vec2.0 models can be fine-tuned for speech recognition with labeled data and a Connectionist Temporal Classification (CTC) loss (Graves et al., 2006). (Baevski et al., 2020) demonstrated that this approach achieved 1.8% word error rate on the test-clean set of Librispeech with a Transformer language model, and 8.3% phone error rate on TIMIT test set without a language model. (Yi et al., 2020) applied wav2vec 2.0 to speech recognition in low-resource languages. The paper reported more then 20% relative improvements in six languages compared with previous work. We have conducted experiments of fine-tuning wav2vec 2.0 with a CTC loss for recognition of suprasegmentals, including syllables, tones, and pitch accents (Yuan et al., 2021). Compared to previous studies, the method achieved 70% error reduction on syllable detection, 50% error reduction on Mandarin tone recognition, and 10% error reduction on pitch accent identification.

2.2. Data Augmentation

Data augmentation is widely used in computer vision (Shorten and Khoshgoftaar, 2019), NLP (Feng et al., 2021), time series (Wen et al., 2020), as well as in speech (Mena et al., 2021). Very briefly, data augmentation methods can be categorized into four different groups: data perturbation, transfer learning, semi-supervised training and generative synthesis. Without loss of generality, let $(x, y) \in \mathcal{D}$ be the input feature and corresponding label of a data sample from training set $\mathcal{D}$. **Data perturbation** does not introduce new data sources, but rather modifies the original x. Common perturbations include adding noise, random cut / crop / rotation / substitution, mixing, etc. **Transfer learning** based techniques try to bring new dataset $\hat{\mathcal{D}}$ to expand $\mathcal{D}$. Although there could be a domain shift, transfer learning methods compensate this by constructing projections from one domain to the other (e.g. adaptors). **Semi-supervised training** solves the problem that part of the y are not gold labels (e.g. closed captions) or even unlabeled. This helps when bringing new data in the same domain but lacking of gold labels. **Generative synthesis** aims to create new data samples (x', y') that is from the same distribution of $\mathcal{D}$. It relies on a generative model such as Generative Adversarial Network (GAN) (Goodfellow et al., 2014), trained on $\mathcal{D}$ or external data sources. We review some popular approaches for speech recognition, from the above 4 different categories.

Data Perturbation: Vocal Tract Length Perturbation (VTLP) (Jaitly and Hinton, 2013) changes each utterance through a warping procedure. (Thai et al., 2019) tries to alter the pitch and speaking rate of the original speech. In (Park et al., 2019), SpecAugment is proposed to mask part of the log mel spectrogram. Mixup technique (Zhang et al., 2018) is adopted in (Meng et al., 2021) to weighted sum the utterances as the augmented speech.

Transfer Learning (out-of-domain data adaption): The recent popularity of pretrain - fine-tune pipelines largely encourage domain adaption. (Hsu et al., 2021) suggests that combining data, both in-domain and out-of-domain, could improve generalization ability during wav2vec2.0 pretraining. This is also verified in an even larger setting (Chan et al., 2021), bigger model and more data. An interesting work in (Fainberg et al., 2016) uses adults' speech to enhance the children's speech recognition, via the out-of-domain stochastic feature mapping (SPF) (Cui et al., 2015) technique.

Semi-supervised Training (bootstraping): This method relies on some seed labeled data for initial supervised training, then generates pseudo labels for other noisy or unlabeled data. The pseudo labels are used to further reinforce the model. This can be done in multiple rounds, and the model can be adjusted using the seed data again (consistency regularization (Xie et al., 2020)) between those rounds. This procedure is termed as bootstraping or self-training in NLP (Yarowsky, 1995), computer vision (Reed et al., 2014) and speech (Punjabi et al., 2019; Chen et al., 2020).

Generative Synthesis: Rather than a simple combination of existing data, generative models learn joint distribution of $p(x, y)$ and sample from it. Variational Autoencoding Wasserstein GAN (VAW-GAN) is used in (Hsu et al., 2017) to build a voice conversion system. Thanks to recent advance of text-to-speech (TTS) systems, a line of works including (Laptev et al., 2020; Rossenbach et al., 2020; Rosenberg et al., 2019), leverage a popular TTS backbone model, Tacotron (Wang et al., 2017), to synthesize new training data. (Tjandra et al., 2017) named such TTS-ASR loop as "machine speech chain mechanism".

Note that the PSST challenge targets the recognition of post-stroke speech. This speech introduces new challenges, as well as opportunities to apply the literature on data augmentation (Geng et al., 2022; Jin et al., 2021; Vachhani et al., 2018) to new scenarios.

2.3. Is More Data Always Better?

In classic machine learning, when the number of data samples N, is less than model capacity (often measured by the number of parameters $|\theta|$), the model tends to overfit due to the bias-variance trade-off (Hastie et al., 2009). However, deep learning models often have a huge amount of parameters that is more than enough to overfit even random labels (Zhang et al., 2021), but such overfitting phenomenon is not commonly seen.

(Belkin et al., 2019) noticed a "double descent" curve, where test loss first becomes worse, then gets better and

better, as the model capacity increases. In (Nakkiran et al., 2021), the authors analyzes double descent curve in deep learning models such as CNNs and Transformers. In particular, they found that within a critical region (the model size falls in a certain range), increasing training data size does not help on testing. But beyond this region (either under-parameterized or over-parameterized cases), more data yields better test performance. (d'Ascoli et al., 2020) even found a "triple descent" phenomenon, and established a connection between model size $|\theta|$, training data size N, and feature dimension d. An asymptotic analysis in (Li et al., 2020) proves that infinite amount of data with infinite dimension could hurt least square estimators' performance.

Rather than simply adding more data, the model could benefit more from improving quality of the added data. For example, analyzing and compensating the domain shift is shown to be very effective in (Gong et al., 2021). In this work, we demonstrate that augmenting from the same domain can significantly improve the PSST recognition results. On the contrary, if augmenting from a different domain, more data may hurt the model's performance.

3. Phone Recognition on TIMIT, Librispeech, and PSST

3.1. Datasets and labels

3.1.1. PSST

The dataset of the PSST challenge (Gale, R., Fleegle, M., Bedrick, S. and Fergadiotis, G., 2022) consists of audio recordings and phonemic transcriptions of people with post-stroke aphasia. The audio data was sourced from the AphasiaBank database (Macwhinney, B., Fromm, D., Forbes, M. and Holland, A., 2011), from which utterances were selected, segmented, and transcribed by experts at Portland Allied Laboratories for Aphasia Technologies (PALAT). The training set contains 2,298 utterances, a total of 2.8 hours of speech. The validation set contains 341 utterances. Additional 652 audio-only utterances were provided for testing, and the results need to be submitted to the organizers for evaluation.

The dataset has 42 labels, including 39 phonemes from the CMU pronouncing dictionary[1], plus /DX/ for flaps, <sil> for long pauses, and <spn> for vocal noises. Excluding <sil> and <spn>, which will be filtered out from evaluation, the size of the label inventory is 40.

3.1.2. TIMIT

TIMIT (Garofolo, J., et al., 1993) has been used as a benchmark dataset for a number of tasks, including phoneme recognition. The corpus contains speech from 630 speakers from different dialect regions of American English, each speaking 10 phonetically balanced sentences. The 6,300 utterances were manually segmented and transcribed at the phone level. Following the literature (Lee and Hon, 1989), the 61 phone labels in the dataset were grouped into 39 categories, representing 38 phonemes plus pause. Compared to PSST, the phoneme /ZH/ does not appear in TIMIT. The corpus also contains a pronouncing dictionary, in which every word has only one canonical pronunciation. Using this dictionary, we generated "canonical" labels for every utterance by simply mapping words into canonical phonemes. The inventory of canonical labels is the same as the inventory of transcribed labels, except for flap, /DX/. Flaps are common in transcriptions (of American English), even though they do not appear in the dictionary.

The TIMIT corpus provides a standard split for training and testing. The training set contains 4,620 utterances (3.9 hours of speech). The remaining 1,680 utterances are in the test set. In our experiments below, we use the test set for validation.

Table 1: Librispeech Splits

Split	Source	Utterances	Hours
Train	train-clean + train-other	281k	960
Validation	dev-clean	2703	
Test	test-clean	2620	

3.1.3. Librispeech

Librispeech (Panayotov, V., Chen, G., Povey, D. and Khudanpur, S., 2015) is a benchmark dataset for English ASR. The corpus is derived from English audiobooks and contains 1000 hours of speech. Unlike TIMIT, LibriSpeech is not phonemically transcribed. It is standard practice to infer canonicalized phonemes. We used g2p-en[2] to convert words into phonemes. The inventory of g2p-en phonemes is the same as those in PSST except for flap, /DX/. Librispeech, when processed by g2p-en, has no flaps.

Librispeech contains subsets called train-clean, train-other, dev-clean, and test-clean. We use train-clean, train-other for training, dev-clean for validation, and test-clean for testing, as reported in Table 1.

.

3.2. PER Within and Across Datasets

We started with the pre-trained model: *wav2vec-vox-new.pt*, a large wav2vec2.0 model trained on the Libri-Light corpus of more than 60k hours of unlabeled speech. We added a linear projection layer to the top of the base model to output phoneme label tokens. The three datasets in Table 2 were used for fine-tuning. The first 10k updates apply to the projection layer, but not the base model. Updates after the first 10k are applied to both the projection layer as well as the Transformer

[1] https://github.com/cmusphinx/cmudict

[2] https://pypi.org/project/g2p-en/

Table 2: Phoneme error rate (PER) and trigram perplexity (per), computed over canonicalized (C) and transcribed (T) phonemes in validation set.

Dataset	C-PER	T-PER	C-per	T-per
TIMIT	1.37%	7.29%	10.9	13.2
Librispeech	1.05%	NA	11.1	NA
PSST	NA	19.4%	NA	10.3

in the base model. Fine-tuning uses a CTC loss. There is a limit of 800k max tokens, which corresponds to 50 seconds of speech at 16k samples per second. The learning rate was 10^{-5}. The metric of unit error rate on the validation set was used to determine the total number of updates. We used *fairseq*[3] for our experiments.

PER is reported in Table 2 for C-phonemes (canonicalized) and T-phonemes (transcribed). Note that C-PER $\ll$ T-PER. The comparison between C-PER and T-PER is easier to make in TIMIT where the gold standard provides both C-phonemes and T-phonemes. These comparisons are more challenging for the other two datasets, where we have one type of phonemes but not the other, and consequently, four cells are NA (not available) in Table 2.

Note that C-PER in Librispeech is relatively close to the C-PER for TIMIT, at about 1% (We also evaluated the Librispeech model on the test set, and the C-PER is 1.12%). The T-PER in PSST and TIMIT are well above 1%. The large differences between C-PER and T-PER are left as an intriguing topic for future research.

Why are T-phonemes so much more difficult than C-phonemes? It is possible that human transcriptions introduce inconsistencies that complicate predictions. Another hypothesis attributes the difference to fine-tuning. It is possible that fine-tuning is learning not only bottom-up acoustic properties of phonemes and contexts (coarticulation), but also top-down constraints (language model). To test this hypothesis, we trained a phoneme trigram language model on the train set, and computed the perplexity of the model on the validation set. As reported in Table 2, the perplexity is larger for transcribed phonemes (T-per $>$ C-per), which may explain in part why recognition of transcribed phonemes is more difficult for wav2vec2.0.

The phone error rate (T-PER) is much higher for PSST. The perplexity of the phoneme language model is, however, similar for PSST, TIMIT and Librispeech. Therefore, it is unlikely that the poor T-PER performance is due to a particular distribution of phonemes in the dataset. In our opinion, factors such as data sparsity, recording conditions, acoustic characteristics of phonemes, and label quality are more likely contributors to the T-PER performance.

We also evaluated the models in a cross-dataset manner. A model trained on one dataset is evaluated on the other datasets (the validation set is used for evaluation). For TIMIT, the model of canonical phonemes was used. The results are listed in Table 3.

[3] https://github.com/pytorch/fairseq

Table 3: Within- and across-dataset PER (within-dataset: validation error; across-dataset: test error.)

	TIMIT	Librispeech	PSST
TIMIT	1.37%	8.48%	39.3%
Librispeech	8.20%	1.05%	34.8%
PSST	14.5%	14.0%	19.4%

Table 4: T-PER for out-of-domain data augmentation. The last column shows performance on PSST (validation split). 3.9 hours of TIMIT (or Librispeech) is better than too much (100+ hours) or too little (none).

Training data			T-PER
In-Domain	TIMIT	Librispeech	PSST
PSST	None	None	19.4%
PSST	3.9 hours	None	**18.0%**
PSST	None	960 hours	30.0%
PSST	None	100 hours	21.6%
PSST	None	3.9 hours	18.7%

Clearly, the models do not transfer well across datasets. The PER of the Librispeech model, for example, is 34.8% on PSST, which is much higher than its within-dataset PER of 1.05%.

Another interesting comparison is along the bottom row of Table 3. Note that $14.5 < 19.4\%$ and $14.0 < 19.4\%$. In other words, the PSST model performed better on TIMIT and Librispeech than on PSST itself.

4. Out-of-Domain Data Augmentation

In this experiment, we supplemented the training data of PSST with training data from TIMIT and Librispeech. For Librispeech, we started with the unabridged training set of 960 hours, but after receiving disappointing results, we repeated the experiment with two smaller samples of 100 hours and 3.9 hours, as shown in Table 4. The choice of 3.9 hours in the last experiment (bottom row of Table 4) was chosen to make the size of the TIMIT training set.

A modest amount of data augmentation is better than too much or too little. That is, the performance of the model was slightly improved when trained with additional data from TIMIT and 3.9 hours of Librispeech. The error rate was decreased from 19.4% of no data augmentation to 18.0% and 18.7%, respectively. On the other hand, the model trained with additional data from the entire train set of Librispeech was significantly degraded with phoneme error rate of 30.0%.

To understand why using more data from Librispeech degrades the model's performance, we plot the contextualized representations of the validation samples from

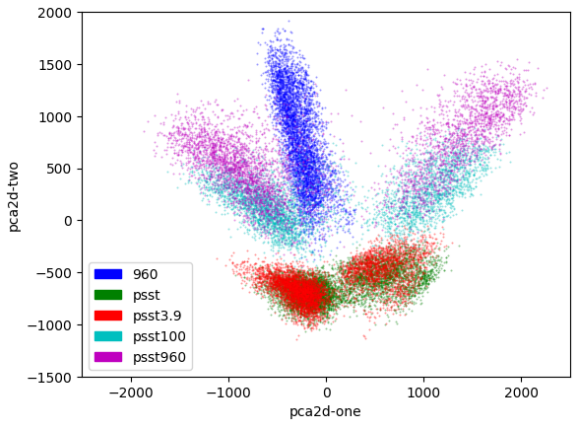

Figure 1: Contextualized representations of PSST validation samples from models trained on different amount of out-of-domain augmentation data, compared with no augmentation (*psst*).

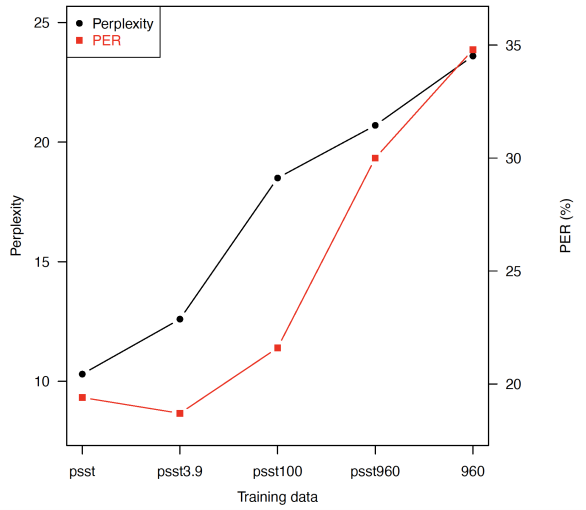

Figure 2: Perplexity of phoneme trigram language model is highly correlated with PER on the validation set of PSST, although language model is not used for decoding.

different models in Figure 1. The contextualized representations were extracted at all frames predicted as a phoneme but not <blank> (i.e., a special token used in CTC). These representations have 1024 dimensions. To make them easier to visualize, we used PCA to project the 1024 dimensions down to 2 dimensions in Figure 1.

Figure 1 shows that *psst* (**green** points) and *psst3.9* (**red** points) occupy similar regions of the plot, in contrast with the three other cases: *960, psst100 psst960*. The **green** points have no training data from Librispeech, and the **red** points have 3.9 hours. The other points have 100+ hours of Librispeech. Augmenting the training data with too much data from Librispeech shifts the representations away from the **green** and **red** points.

As discussed above, the contextualized representations form a finetuned wav2vec2.0 may contain language model information besides phonetic properties. The shift of the representations by out-of-domain data may suggest a mismatch in language model between "in" and "out" domains. To test this hypothesis, we trained a phoneme trigram language model for each amount of augmentation data, and computed the perplexity of the model on the validation set of PSST. The results are shown in Figure 2.

Figure 2 shows that perplexity increases from left to right. The large differences in perplexity indicate large differences in domains. The language model for Librispeech is very different from the language model for PSST. Increases in perplexity tend to degrade performance (in terms of PER). That is, adding too much data from Librispeech tends to increase PER.

However, *psst3.9* is an important exception. In this case, adding 3.9 hours of out-of-domain data increases perplexity by a modest amount. However, PER moves in the opposite direction. We susspect that improvements in phonetic representations are large enough to more than compensate for the modest increase in perplexity. Thus, adding 3.9 hours of Librispeech is better (in terms of PER) than too much (100+ hours) and too little (none).

5. In-Domain Data Augmentation

5.1. Extracting Utterances from AphasiaBank

The PSST dataset was derived from AphasiaBank. In this experiment we extracted 48,937 utterances (47 hours) from aphasia subjects in AphasiaBank, excluding recording sessions that include samples in the test set. Because only word transcription is available, we tried two methods to use these utterances for phoneme recognition. The first method is to use audio only for semi-supervised training. The second method is to obtain phoneme labels from word transcription through forced alignment.

In the first method we used the model trained on the train set of PSST to predict phonemes (i.e., pseudo labels). For each utterance, we also computed a confidence/probability score by averaging the probabilities of 1-best hypothesis at frames where the prediction is a phoneme but not <blank>. The distribution of the probability scores are shown in Figure 3. The probability score will be used to either select utterances or weight a CTC loss, as described below.

In the second method, we employed the Penn Phonetics Lab Forced Aligner (P2FA) to do forced alignment (Yuan and Liberman, 2008). More than half of the extracted utterances cannot be easily aligned because the transcription contains OOVs (out-of-vocabulary), e.g., "*xxx*". Only utterances with "clean" word transcription, 22,836 out of 48.937, were aligned to get phoneme labels for training.

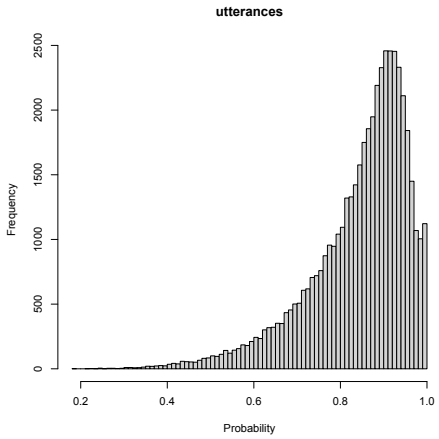

Figure 3: Distribution of AphasiaBank-utterance probability scores from a model trained on PSST.

5.2. Filtering on Confidence and Weighted/Unweighted CTC Loss

To use predicted phonemes or pseudo labels, we define a threshold to select utterances for which the model has more confidence in its prediction. We experimented with five thresholds: 0, 0.7, 0.8, 0.9, and 0.95. When the threshold is 0, all utterances are selected. Higher thresholds filter out more utterances with less confidence. The selected utterances were added to the PSST training set in one of two conditions: (1) weighted or (2) unweighted. The unweighted condition uses standard CTC loss. This condition treats AphasiaBank utterances equally with utterances in the PSST training set. In contrast, the weighted condition uses Eq. (1) to compute CTC loss in the fine-tuning process.

$$\mathcal{L}_{\text{CTC}} = \frac{1}{|\mathcal{B}|} \sum_{(x,y,s)\in\mathcal{B}} -s \log P(y|x) \qquad (1)$$

Eq. (1) computes CTC loss, $\mathcal{L}_{\text{CTC}}$, for a batch of $\mathcal{B}$ samples. Each sample consists of input frames, x, and a label, y, with a probability score, s. When training on pseudo-labels, y is a pseudo-label and s is a score from the system, where $0 \leq s \leq 1$. When training on ground truth labels from PSST, y is a label from the gold standard, and $s = 1$.

5.3. Results

Table 5 reports results on the validation set of PSST for a number of thresholds, with and without weighting. The last row reports results for forced alignment (FA). Data augmentation improves over the baseline in all conditions, with an absolute error reduction between 1.3% and 2.1%. Weighting is helpful when the threshold is small, but the differences between weighted CTC and unweighted CTC diminish for larger thresholds. PER performance improves if we exclude "bad" utterances (or downweight them). PER is 18.1% for all utterances, and reduces to 17.3% with a threshold of 0.9. This threshold selects 18k (of 48k) utterances.

Table 5: PER of in-domain data augmentation using different thresholds and CTC weighting. Baseline PER is 19.4%, and FA (forced alignment) PER is 17.9%.

Threshold	Utterances	Unweighted CTC	Weighted CTC
0	48,497	18.1%	18.0%
0.7	42,816	17.9%	17.4%
0.8	35,096	17.4%	-
0.9	18,296	17.3%	17.4%
0.95	6488	17.9%	-
FA	22,836	17.9%	-

Table 6: PER at each iteration of data augmentation, with the number of selected utterances in parentheses.

Iteration	Unweighted CTC	Weighted CTC
1	17.3% (18,296)	17.4% (42,816)
2	16.9% (28,188)	16.9% (43,793)
3	16.7% (33,554)	16.8% (46,223)

5.3.1. Results with Iteration

After a new model was trained with data augmentation, we used it to predict phonemes for utterances extracted from AphasiaBank. The predictions and probability scores are different from predictions without data augmentation. We use the new predictions and scores to select a new set of utterances. We iterated this procedure until no further improvement could be made. Table 6 reports results for a number of thresholds, with and without CTC weighting. The Table shows that our best model achieved 16.7% phone error rate on the validation set of PSST, representing a relative error reduction of 14% over the baseline model trained without data augmentation.

Figure 4 shows contextualized representations (2-D PCA projections) of the best model **red** and the baseline model **green**. Note that the **red** and **green** points occupy similar regions of the plot, unlike models of out-of-domain augmentation shown in Figure 1.

6. Conclusions

Fine-tuning wav2vec2.0 with a CTC loss not only learns bottom-up acoustic properties but also top-down constraints. In the task of phoneme recognition, a phoneme language model is implicitly learned from fine-tuning and represented in a fine-tuned model. Therefore, for the method of fine-tuning wav2vec2.0, out-of-domain data augmentation is likely to degrade performance if there is a language-model mismatch between "in" and "out" domains. Our study confirms this

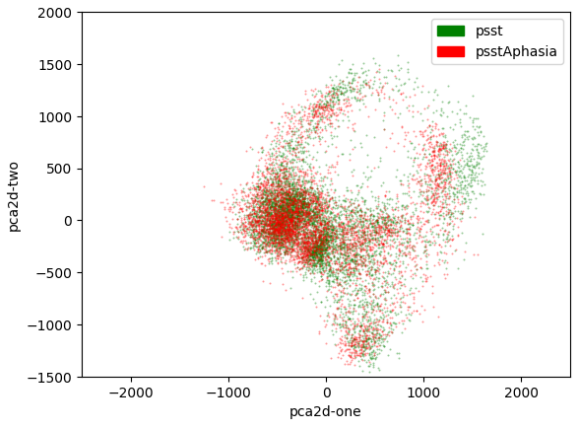

Figure 4: Contextualized representations of PSST validation samples from models trained with (*psstAphasia*) and without (*psst*) in-domain data augmentation.

hypothesis. We found that although a modest amount of out-of-domain data helps phoneme recognition from speakers with aphasia, too much out-of-domain data will degrade performance. Visualizations showed that out-of-domain data augmentation shifts the space of representations learned from fine-tuning away from the corresponding space for a baseline model. Visualizations also showed that in-domain data augmentation does not shift the space as much as out-of-domain data augmentation.

It is difficult to obtain large quantities of speech with phonemic transcriptions from subjects with aphasia. We extracted audio utterances from AphasiaBank and generated predictions (pseudo labels) from a baseline model, and used this resource for in-domain data augmentation. We found that excluding utterances with less confident predictions can lead to a better performance of the model. Therefore, for both out-of-domain and in-domain data augmentation, we found scenarios where "less is more".

We iterated the procedure of in-domain data augmentation by training a new model and updating predictions and confidence scores with the new model, until convergence. Our final model achieved 16.7% phone error rate on the PSST validation set, without using a language model for decoding. This result represents a relative error reduction of 14% over the baseline model trained without data augmentation. The results on the test set were submitted to the challenge for evaluation.

Finally, we found that with the method of fine-tuning wav2vec2.0 "canonicalized" phonemes are much easier to recognize than manually transcribed phonemes. On TIMIT, the phoneme error rate was 1.37% and 7.29% respectively for the two types of labels. On Librispeech, the phoneme error rate of "canonicalized" phonemes reached as low as 1.05%. This is an intriguing result. More research is needed, from both linguistics and machine learning, to fully understand it.

7. Bibliographical References

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Chan, W., Park, D., Lee, C., Zhang, Y., Le, Q., and Norouzi, M. (2021). Speechstew: Simply mix all available speech recognition data to train one large neural network. *arXiv preprint arXiv:2104.02133*.

Chen, Y., Wang, W., and Wang, C. (2020). Semi-supervised asr by end-to-end self-training. *arXiv preprint arXiv:2001.09128*.

Cui, X., Goel, V., and Kingsbury, B. (2015). Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9):1469–1477.

d'Ascoli, S., Sagun, L., and Biroli, G. (2020). Triple descent and the two kinds of overfitting: Where & why do they appear? *Advances in Neural Information Processing Systems*, 33:3058–3069.

Fainberg, J., Bell, P., Lincoln, M., and Renals, S. (2016). Improving children's speech recognition through out-of-domain data augmentation. In *Interspeech*, pages 1598–1602.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Foygel, D. and Dell, G. (2000). Models of impaired lexical access in speech production. *Journal of Memory and Language*, 43:182–216, 08.

Geng, M., Xie, X., Liu, S., Yu, J., Hu, S., Liu, X., and Meng, H. (2022). Investigation of data augmentation techniques for disordered speech recognition. *arXiv preprint arXiv:2201.05562*.

Gong, C., Wang, D., Li, M., Chandra, V., and Liu, Q. (2021). Keepaugment: A simple information-preserving data augmentation approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1055–1064.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Hastie, T., Tibshirani, R., Friedman, J. H., and Fried-

man, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hsu, C.-C., Hwang, H.-T., Wu, Y.-C., Tsao, Y., and Wang, H.-M. (2017). Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. *arXiv preprint arXiv:1704.00849*.

Hsu, W.-N., Sriram, A., Baevski, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., et al. (2021). Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. *arXiv preprint arXiv:2104.01027*.

Jaitly, N. and Hinton, G. E. (2013). Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117, page 21.

Jin, Z., Geng, M., Xie, X., Yu, J., Liu, S., Liu, X., and Meng, H. (2021). Adversarial data augmentation for disordered speech recognition. *arXiv preprint arXiv:2108.00899*.

Laptev, A., Korostik, R., Svischev, A., Andrusenko, A., Medennikov, I., and Rybin, S. (2020). You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation. In *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 439–444. IEEE.

Lee, K.-F. and Hon, H.-W. (1989). Speaker-independent phone recognition using hidden markov models. *IEEE Trans. Acoust. Speech Signal Process.*, 37:1641–1648.

Li, Z., Xie, C., and Wang, Q. (2020). Provable more data hurt in high dimensional least squares estimator. *arXiv preprint arXiv:2008.06296*.

Ma, D., Ryant, N., and Liberman, M. (2021). Probing acoustic representations for phonetic properties. *Proceedings of ICASSP 2021*.

Mena, C., DeMarco, A., Borg, C., van der Plas, L., and Gatt, A. (2021). Data augmentation for speech recognition in maltese: A low-resource perspective. *arXiv preprint arXiv:2111.07793*.

Meng, L., Xu, J., Tan, X., Wang, J., Qin, T., and Xu, B. (2021). Mixspeech: Data augmentation for low-resource automatic speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7008–7012. IEEE.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2021). Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

Punjabi, S., Arsikere, H., and Garimella, S. (2019). Language model bootstrapping using neural machine translation for conversational speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 487–493. IEEE.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. (2014). Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

Rosenberg, A., Zhang, Y., Ramabhadran, B., Jia, Y., Moreno, P., Wu, Y., and Wu, Z. (2019). Speech recognition with augmented synthesized speech. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 996–1002. IEEE.

Rossenbach, N., Zeyer, A., Schlüter, R., and Ney, H. (2020). Generating synthetic audio data for attention-based speech recognition systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7069–7073. IEEE.

Shah, J., Singla, Y. K., Chen, C., and Shah, R. R. (2021). What all do audio transformer models hear? probing acoustic representations for language delivery and its structure. *ArXiv:2101.00387*.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

Thai, B., Jimerson, R., Arcoraci, D., Prud'hommeaux, E., and Ptucha, R. (2019). Synthetic data augmentation for improving low-resource asr. In *2019 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, pages 1–9. IEEE.

Tjandra, A., Sakti, S., and Nakamura, S. (2017). Listening while speaking: Speech chain by deep learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 301–308. IEEE.

Vachhani, B., Bhat, C., and Kopparapu, S. K. (2018). Data augmentation using healthy speech for dysarthric speech recognition. In *Interspeech*, pages 471–475.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., et al. (2017). Tacotron: Towards end-to-end speech synthesis. *Proc. Interspeech 2017*, pages 4006–4010.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., and Xu, H. (2020). Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*.

Wilson, S., Henry, M., Besbris, M., Ogar, J., Dronkers, N., Jarrold, W., Miller, B., and Gorno-Tempini, M. L. (2010). Connected speech production in three variants of primary progressive aphasia. *Brain : a journal of neurology*, 133:2069–88, 07.

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. (2020). Unsupervised data augmentation for consis-

tency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Yi, C., Wang, J., Cheng, N., Zhou, S., and Xu, B. (2020). Applying wav2vec2.0 to speech recognition in various low-resource languages. *ArXiv:2012.12121*.

Yuan, J. and Liberman, M. (2008). Speaker identification on the scotus corpus. *ournal of the Acoustical Society of America*, 123:3878.

Yuan, J., Ryant, N., Cai, X., Church, K., and Liberman, M. (2021). Automatic recognition of suprasegmentals in speech. *ArXiv:2108.01122*.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

8. Language Resource References

Gale, R., Fleegle, M., Bedrick, S. and Fergadiotis, G. (2022). *Dataset and tools for the PSST Challenge on Post-Stroke Speech Transcription*. https://doi.org/10.5281/zenodo.6326002.

Garofolo, J., et al. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus (LDC93S1)*. https://catalog.ldc.upenn.edu/LDC93s1.

Macwhinney, B., Fromm, D., Forbes, M. and Holland, A. (2011). *AphasiaBank: Methods for Studying Discourse*. https://doi.org/10.1080/02687038.2011.589893.

Panayotov, V., Chen, G., Povey, D. and Khudanpur, S. (2015). *Librispeech: an ASR corpus based on public domain audio books*. https://www.openslr.org/12.