SYNTHETIC DATA AUGMENTATION FOR IMPROVING LOW-RESOURCE ASR

Bao Thai[1], *Robert Jimerson*[1], *Dominic Arcoraci*[1], *Emily Prud'hommeaux*[1,2], *Raymond Ptucha*[1]

[1]Rochester Institute of Technology, Rochester, New York, USA
[2]Boston College, Chestnut Hill, Massachusetts, USA

ABSTRACT

Although the application of deep learning to automatic speech recognition (ASR) has resulted in dramatic reductions in word error rate for languages with abundant training data, ASR for languages with few resources has yet to benefit from deep learning to the same extent. In this paper, we investigate various methods of acoustic modeling and data augmentation with the goal of improving the accuracy of a deep learning ASR framework for a low-resource language with a high baseline word error rate. We compare several methods of generating synthetic acoustic training data via voice transformation and signal distortion, and we explore several strategies for integrating this data into the acoustic training pipeline. We evaluate our methods on an indigenous language of North America with minimal training resources. We show that training initially via transfer learning from an existing high-resource language acoustic model, refining weights using a heavily concentrated synthetic dataset, and finally fine-tuning to the target language using limited synthetic data reduces WER by 15% over just transfer learning using deep recurrent methods. Further, we show improvements over traditional frameworks by 19% using a similar multistage training with deep convolutional approaches.

Index Terms— speech recognition, deep learning, data augmentation, low-resource languages

1. INTRODUCTION

The use of deep neural networks in acoustic modeling for automatic speech recognition (ASR) has resulted in remarkable accuracy gains for English, Mandarin, and other high-resource languages [1, 2, 3, 4, 5]. These methods, however, require very large amounts of training data in order to yield improvements on this scale. Deep learning ASR systems for languages with very limited training resources typically must incorporate additional training resources, such as cross-lingual acoustic models or in-domain synthetic acoustic data, to begin to approach the word error rates found using traditional hidden Markov model (HMM) and Gaussian mixture model (GMM) ASR frameworks.

Some of the most successful methods for adapting deep learning ASR to low-resource scenarios have relied on cross-lingual or multilingual transfer learning, in which a pre-trained acoustic model for a language with abundant training data is used to initialize the weights of the acoustic model for a language with limited data [6, 7, 8].

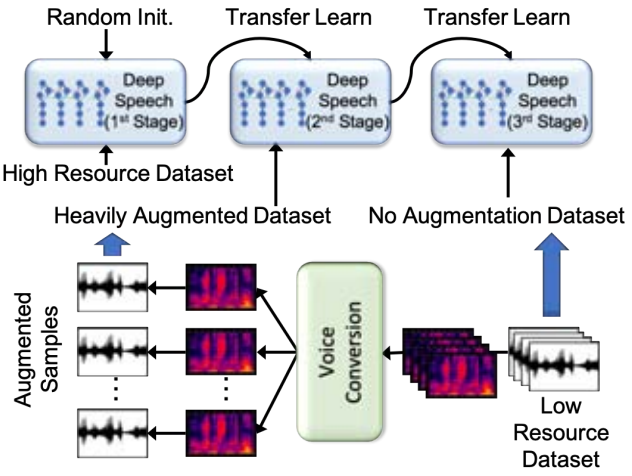

Fig. 1. Multistage training of DeepSpeech using transfer learning with heavily augmented samples followed by retraining with no augmentation.

In this paper, we explore several methods of synthetic acoustic data augmentation within both recurrent and convolutional deep learning ASR frameworks for a low-resource language in order to complement and supplement the use of transfer learning. Previous work has shown that simple augmentation methods relying on speech signal distortion (e.g., shifting F0, adding background noise) results in WER reductions for resource constrained ASR systems [9, 10, 11, 12, 8]. Here, we compare simple data augmentation techniques to more complex approaches in which new synthetic audio data is generated from the existing acoustic training data using voice conversion. We use a very small set of transcribed recordings (10 hours) of the Seneca language, a critically endangered, morphologically complex, low-resource language of North America. We investigate novel ways of incorporating this synthetic data into both recurrent and convolutional acoustic training pipelines. We find that some, but not all, of our augmentation methods yield substantial improvements in

978-1-7281-4352-1/19/$31.00

ASR accuracy over two baseline deep ASR models that do not rely on data augmentation. We also find that convolutional approaches have both computational and accuracy advantages over their recurrent counterparts. Our work demonstrates that even languages with scarce training resources can benefit from the use of deep learning ASR methods.

The main contributions of this work are: 1) A novel deep learning approach for resource-constrained ASR using multiple stage learning; 2) A demonstration that deep learning ASR methods can achieve results superior to those of traditional HMM/GMM methods, even in the absence of abundant training data; and 3) A transcribed corpus of 12 hours of the Seneca language, a critically endangered language spoken in Canada and New York State.

2. BACKGROUND

Given sufficient in-domain monolingual training data, deep neural network methods for ASR generally outperform traditional methods relying on HMMs and GMMs, often by very wide margins [1, 13, 2, 14, 3, 15, 16, 5, 17]. These approaches typically use RNNs to convert variable length audio wave segments or spectrograms to text at the phone or character level. Methods that produce characters, such as DeepSpeech [14], currently use Connectionist Temporal Classification (CTC) to reduce streams of characters to plausible words. These words are then optionally passed through a language model to yield sequences of attested words. Because of the limitations of CTC, Battenberg et al. [15] and others utilize sequence-to-sequence models which first encode an entire sequence, then decode one character at a time until the end of the utterance.

Despite deep RNNs success in ASR and other sequence modelling tasks, these networks cannot easily take advantage of parallelization on modern hardware since the output of an RNN cell at each timestep depends on the operation from the previous timestep, leading to longer training times. Rather than use recurrent layers, there have been several architectures relying on convolution [18, 19, 20] to capture temporal dependencies. These techniques improve the training speed of sequence-related tasks while still delivering competitive performance. In particular, Liptchinsky et al. [20] achieves the same performance on the LibriSpeech clean test set as the best recurrent model ([14]) despite using an order of magnitude less data.

Some early approaches for adapting deep learning models to low-resource scenarios focused on changes to the model architecture [21, 9, 6], but greater success was often found in model adaptation in the form of transfer learning from a model built using data from one or more resource-rich languages [9, 22, 23]. The introduction of synthetic data into the training corpus has also been found to yield improvements in true low-resource, artificially low-resource, and resource-rich conditions [9, 10, 11, 12]. Jimerson et al. [8] showed that simple augmentation methods such as adding noise, modifying f0, changing speaking rate, and other distortions resulted in WER reductions for an ASR system trained on less than two hours of audio. Similar techniques have also been used to improve dysarthric speech recognition [24].

Another approach to augmenting the acoustic training data is to synthesize new versions of existing data using voice conversion. Much of the recent work in this area derives from work in generating synthetic images. Generative Adversarial Network (GAN) [25] architectures have proven successful in generating synthetic images that mimic the distribution of input data [25, 26, 27, 28]. Chang et al. [27] used GANs along with reinforcement learning to generate augmented sentences with code switching between English and Taiwanese. Choi et al. [28] perform many-to-many style transformations of input images from one domain to another while keeping other characteristics, such as object shape, the same. Kameoka et al. [29] applied the architecture proposed in [28] to acoustic features to perform many-to-many voice conversion in an architecture called StarGAN-VC. Since StarGAN-VC requires only a few minutes of non-parallel, unlabeled speeches from each speaker, we find the architecture a good fit for low-resourced language voice conversion.

Hsu et al. [30] proposed a different method to perform non-parallel voice conversion by combining a variational autoencoder with a Wasserstein GAN. This architecture (VAW-GAN) uses a conditional variational autoencoder (C-VAE) to model the acoustic features of speech from each speaker and a Wasserstein GAN to synthesize speeches from a different speaker. By using a Wasserstein GAN instead of a vanilla GAN, this architecture produces sharper, more structured spectral envelopes compared to similar methods discussed in Hsu et al. [31], among others.

3. DATA

We perform our experiments on Seneca, a morphologically complex and endangered language of North America, spoken as a first language by roughly 50 individuals and as a second language by a few hundred others. The available audio recordings consist of roughly 720 minutes of spontaneous, naturalistic speech produced by eleven adult speakers, eight male and three female. All speakers are first-language Seneca speakers whose second language is English, and all eleven are middle-aged or elderly. Recordings were made over many years under a variety of conditions, yielding a diverse range of audio quality. Details of the training data can be found in the Supplementary Material.

4. METHOD

4.1. Acoustic modeling

We built several different baseline models using three different ASR frameworks. One model uses the GMM/HMM

framework implemented in the Kaldi[1] toolkit; two models use the DNN-RNN framework of DeepSpeech[2] [2]; and two models use a simplified 1D gated convolutional neural network from a model proposed by Liptchinsky et al. [20].

Kaldi: The first baseline model ("Baseline: Kaldi") uses the traditional HMM/GMM framework provided by Kaldi, with a triphone model, a trigram KenLM [32] language model, and the parameter settings recommended in the Kaldi tutorial.

DeepSpeech: The DeepSpeech model consists of a five-layer recurrent neural network with Long-Short Term Memory (LSTM) cells. The first, second, third and fifth layers of the neural network are fully connected, while the fourth layer is a bi-directional recurrent layer. All layers contain 2048 hidden units and are followed by a dropout layer of 0.2. Raw audio input is partitioned into windows of length 20msec strided by 10msec. These raw inputs are converted to 13 MFCCs, along with the deltas and delta-deltas, in preparation for the first layer of the neural network. To train the model, CTC loss [33] was used.

mini-GCNN: The mini-GCNN architecture we used (Supplementary Material, Fig. 2a) was a modification from the model described in Liptchinsky et al. [20]. The original model consists of 17 Gated Linear Unit (GLU) blocks, with kernel size increasing from 3 to 21 and number of filters increasing from 100 to 375. These GLU blocks are followed by a 1×1 GLU block with a depth of 1000 to mimic a fully-connected layer. To obtain character probability at each timestep, another 1×1 convolution layer is used, followed by log-softmax activation function. To speed up training time and simplify the network to suit the volume of data available, we removed GLU blocks with kernel size greater than 8 but kept the block with kernel size of 21 to retain the ability to capture long temporal dependency. Additionally, we added a GLU block with kernel size of 13 before the GLU block with kernel size of 3 to improve medium-range temporal dependency. More details are in the Supplimentary Material.

Within DeepSpeech and mini-GCNN frameworks, we establish two baseline models for each architecture. The first models ("Baseline: DeepSpeech" and "Baseline: mini-GCNN") were trained using random weight initialization and the unaugmented 600 minutes of Seneca data. Using transfer learning, the second models ("Baseline: DeepSpeech w/TL" and "Baseline mini-GCNN w/TL") were initialized using weights from a model trained on English only, using the 960-hour LibriSpeech corpus [34]. The final fully-connected layer in DeepSpeech and the final convolution layer in mini-GCNN were replaced to match the number of tokens in the Seneca alphabet. The models with transfer learning were then trained on the unaugmented 600 minutes of Seneca data.

[1]http://kaldi-asr.org/doc/

[2]https://github.com/mozilla/DeepSpeech/tree/v0.3.0

4.2. Multistage transfer learning

We implement a multistage transfer learning strategy in conjunction with data augmentation as described in the Supplementary Materials Section 8.3. The following training procedure is carried out for each of the three data augmentation methods. Figure 1 shows the overall architecture.

In the first stage, a DeepSpeech or mini-GCNN model is trained on 960-hour of LibriSpeech English corpus, with randomized initial weights. The DeepSpeech model was trained with learning rate of 0.001, and the mini-GCNN model was trained with learning rate of 0.0003. The models are trained for 75 epochs, with the best models saved to initialize weights in later stage. The best models were determined by word-error rate on the LibriSpeech validation set.

In the second stage, the weights of a second DeepSpeech or mini-GCNN model are initialized using the weights from the best English models obtained in the first stage. The training data from this second model includes the original unaltered 10-hour Seneca dataset as well as up to a $10\times$ augmentation of each utterance using one of the three data augmentation methods: speed and pitch modification (Augment10), voice conversion via StarGAN-VC, or voice conversion via VAWGAN. This model is trained until convergence on the training dataset. For DeepSpeech, we used learning rate of 0.001. For the mini-GCNN model, we used learning rate of 0.0003.

The motivation for the final stage is that the augmented data often contains heavy digital artifacts. Since the amount of augmented data is significantly higher than the amount of original data, the networks trained on augmented and original data might be skewed towards improving performance on augmented data. However, it is hoped that the network can still learn valuable representation with augmented data, which will allow the final network trained on original data to perform better.

In the final stage, the weights of the third DeepSpeech or mini-GCNN model are initialized using the final weights from the second model. The training data for the third models includes only the original 10 hours of unaltered Seneca dataset with no augmented data. Models in this fine-tuning stage are trained until convergence on the training dataset. For DeepSpeech, we used 0.0001 learning rate. For mini-GCNN, we used 0.00003 learning rate.

5. RESULTS

Table 1 shows the performance across all methods evaluating using both word error rate (WER) and character error rate (CER). For each acoustic model, we evaluate with: (1) no language model (NO LM); (2) the trigram word-level language model described in Supplementary Material (W/LM). We replace all WER values greater than 1.0 with the value 1.0 indicating that little or no correct output was produced.

	DS (NO LM)		DS (W/LM)		m-GCNN (NO LM)		m-GCNN (W/LM)	
	WER	CER	WER	CER	WER	CER	WER	CER
Baseline: no TL, no Aug	1.000	0.891	0.970	0.872	0.839	0.365	0.426	0.257
Baseline: + TL, no Aug	0.859	0.436	0.727	0.409	0.766	0.299	0.350	0.186
Augment10 Stage2	1.000	0.716	0.975	0.698	0.702	0.266	0.372	0.184
Augment10 Stage3	0.850	0.427	0.693	0.421	**0.686**	**0.256**	0.364	0.179
VAWGAN Stage 2	1.000	0.753	1.174	0.702	0.750	0.296	0.381	0.207
VAWGAN Stage 3	0.904	0.468	0.800	0.444	0.710	0.271	0.354	0.177
StarGAN-VC Stage 2	0.911	0.497	0.790	0.474	0.817	0.364	0.488	0.311
StarGAN-VC Stage 3	**0.722**	**0.364**	**0.571**	**0.333**	0.691	0.263	**0.343**	**0.173**

Table 1. WER (word error rate) and CER (character error rate) for various transfer learning (TL) and augmentation strategies (rows) vs. DeepSpeech (DS) and mini-GCNN (m-GCNN) with and without a trigram language model. Kaldi with the same language model gives a WER/CER of 0.530/0.307.

We see that for DeepSpeech, the baseline model using only the 10 hours of unaugmented Seneca data yields a WER greater than 1.0, meaning it yielded little or no correct output. Applying a language model slightly reduces both WER and CER. With transfer learning from an acoustic model pre-trained on English data, we see more promising results, particularly when a language model is applied. All Stage 2 models, which were trained with both augmented and unaugmented data, produce error rates similar to those of the Seneca-only baseline. The most promising DeepSpeech results are those of the Stage 3 data augmentation models, particularly StarGAN-VC and Augment10, both of which improve substantially over the baseline model using transfer learning, especially when a language model is used during decoding. We see that the StarGAN-VC model yields a 40-point improvement over the Seneca-only baseline DeepSpeech model and a 15-point improvement over the transfer learning baseline model. While this model approaches the Kaldi model, it still does not perform as well as the traditional GMM/HMM approach.

The baseline mini-GCNN acoustic model, trained on only the 10 hours of Seneca data with no transfer learning or data augmentation, yields a substantially lower WER than that achieved by the baseline DeepSpeech model. Decoding with the n-gram language model results in a further 10-point improvement in WER over the baseline Kaldi model. With transfer learning from a pre-trained English model and a language model, we see a additional 7.6-point improvement in WER. While all augmentation methods show improvement over Kaldi model and the baseline mini-GCNN model without transfer learning, we do not see as large reductions in WER from data augmentation with mini-GCNN as we do with DeepSpeech. The stage 3 mini-GCNN model with StarGAN-VC augmentation shows the best WER at 0.343.

The superiority of mini-GCNN over DeepSpeech might be attributable to the smaller number of parameters in the model miniGCNN model. The DeepSpeech model has over 45 million parameters, compared to 4 million parameters in the mini-GCNN model. While a model with more parameters can learn more complex functions and relationships, training such a model requires more data. Additionally, the GLU blocks in mini-GCNN allows the model to learn the importance of each feature map, which improves the gradient flow as well as the ability of the model to extract useful information from each block.

6. CONCLUSIONS

In this paper, we proposed a multistage deep learning approach for low-resource ASR which uses both transfer learning and data augmentation via speech signal distortion and voice conversion. We show that transfer learning and data augmentation independently contribute to meaningful reductions in word error rate. In addition, the weak results after the second stage of training with augmented data indicate that the final fine-tuning stage, in which augmented models are retrained using only unaugmented data, is a crucial component of the training procedure.

7. REFERENCES

[1] Geoffrey Hinton, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Brian Kingsbury, and Tara Sainath, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, November 2012.

[2] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[3] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,"

in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[4] Kartik Audhkhasi, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Michael Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4759–4763.

[5] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[6] Dongpeng Chen, Brian Mak, Cheung-Chi Leung, and Sunil Sivadas, "Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5592–5596.

[7] Hardik Sailor, Ankur Patil, and Hemant Patil, "Advances in low resource asr: A deep learning perspective," in *Proceedings of the 2018 Workshop on Spoken Language Technology for Under-resourced Languages*, 08 2018, pp. 162–166.

[8] Robbie Jimerson, Kruthika Simha, Raymond Ptucha, and Emily Prudhommeaux, "Improving asr output for endangered language documentation," in *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, pp. 182–186.

[9] Mark JF Gales, Kate M Knill, Anton Ragni, and Shakti P Rath, "Speech recognition and keyword spotting for low-resource languages: Babel project research at CUED.," in *SLTU*, 2014, pp. 16–23.

[10] Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney, "Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[11] Jayadev Billa, "Isi asr system for the low resource speech recognition challenge for indian languages," *Proc. Interspeech 2018*, pp. 3207–3211, 2018.

[12] Matthew Wiesner, Adithya Renduchintala, Shinji Watanabe, Chunxi Liu, Najim Dehak, and Sanjeev Khudanpur, "Low resource multi-modal data augmentation for end-to-end asr," 2018.

[13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing ICASSP, 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[14] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, and Guoliang Chen, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[15] Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu, "Exploring neural transducers for end-to-end speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 206–213.

[16] Yu Zhang, William Chan, and Navdeep Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4845–4849.

[17] Wiehan Agenbag and Thomas Niesler, "Automatic subword unit discovery and pronunciation lexicon induction for asr with application to under-resourced languages," *Computer Speech & Language*, vol. 57, pp. 20–40, 2019.

[18] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[19] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 933–941.

[20] Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert, "Letter-based speech recognition with gated convnets," *CoRR*, vol. abs/1712.09444, 2017.

[21] Yajie Miao, Florian Metze, and Shourabh Rawat, "Deep maxout networks for low-resource speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 398–403.

[22] Frantisek Grézl, Martin Karafiát, and Karel Vesely, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7654–7658.

[23] David Imseng, Petr Motlicek, Hervé Bourlard, and Philip N Garner, "Using out-of-language data to improve an under-resourced speech recognizer," *Speech communication*, vol. 56, pp. 142–151, 2014.

[24] Bhavik Vachhani, Chitralekha Bhat, and Sunil Kumar Kopparapu, "Data augmentation using healthy speech for dysarthric speech recognition," in *Proceedings of Interspeech*, 09 2018, pp. 471–475.

[25] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial networks," 2014.

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017.

[27] Ching-Ting Chang, Shun-Po Chuang, and Hung-Yi Lee, "Code-switching sentence generation by generative adversarial networks and its application to data augmentation," *arXiv preprint arXiv:1811.02356*, 2018.

[28] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," *CoRR*, vol. abs/1711.09020, 2017.

[29] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion with star generative adversarial networks," *arXiv preprint arXiv:1806.02169*, 2018.

[30] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," *CoRR*, vol. abs/1704.00849, 2017.

[31] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.

[32] Kenneth Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the sixth workshop on statistical machine translation*. Association for Computational Linguistics, 2011, pp. 187–197.

[33] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

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

8. SUPPLEMENTARY MATERIAL

8.1. Seneca dataset details

Twelve hours of recordings were transcribed using Seneca's current orthographic conventions and segmented at the utterance level by second-language Seneca speakers. Because Seneca orthography is quite reliably phonemic, with few ambiguous character-to-phone and phone-to-character mappings, we choose to treat characters as phones.

The transcribed audio data was partitioned into a 10-hour training set and a 2-hour test set as follows. Using the utterance boundaries provided in the reference transcripts, we randomly selected individual utterances from the full corpus of twelve hours until we had compiled ten hours of audio for training. The remaining two hours comprise the test set. We deliberately selected utterances in a random fashion in order to maximize diversity of gender, age, dialect, voice quality, and content (e.g., narrative vs. conversation) of both the training and test sets and to avoid overfitting to any particular speaker or speaker characteristic. We note that selecting the test data in this way has the effect that certain speakers appear in both the testing and training data, a compromise we are obliged to make given the very small number of available speakers of the language.

Approximately six hours of the audio data consists of casual conversations between one of the authors and a Seneca elder dealing with current events, the weather, and anecdotes from the elder's childhood. The remaining audio data was collected from a variety of other speakers and consists of community narratives and information about the natural world. In addition to transcriptions of this audio (roughly 35,000 words), the text data used to train the language model includes an additional 6000 words of previously transcribed texts for which there are no corresponding audio recordings.

8.2. Mini-GCNN details

Fig. 2b shows the architecture of each GLU block. Each block consists of two 1D convolutions of the same kernel size and depth. One of the convolution outputs (Conv_A) is used to extract features from the input, while the other convolution output (Conv_B) is passed through a sigmoid activation function and then multiplied element-wise to act as a gating mechanism. This gating mechanism allows the network to learn which feature is important and should be passed on to the next layer, which help improve gradient flow. Dropout layers of 0.25 is used after each GLU block for regularization. While the original GCNN network is trained with MFSC as input, we decided to go with MFCC instead. Similar to the DeepSpeech model, the input MFCCs are obtained from window size of 20msec and strided by 10msec, and CTC loss was used to train the network.

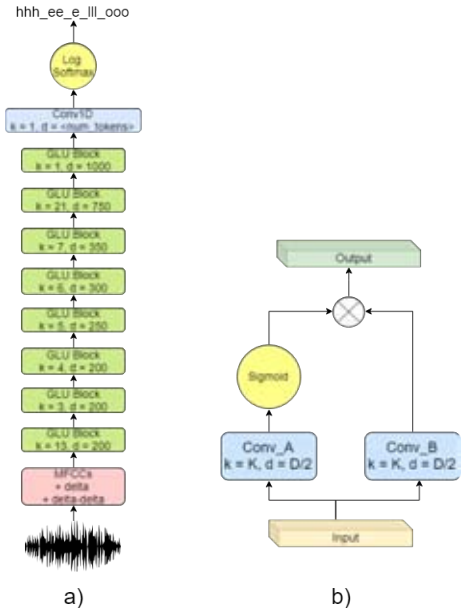

Fig. 2. a) The mini-GCNN architecture used was modified from the architecture used for WSJ described in [20]. We removed layers with kernel size greater than 8 to simplify the network and speed up training. For each block, k is the kernel size, and d is the depth given to the GLU. b) The architecture of each GLU block, where k is kernel size, d is convolution depth of an operation, and D is the convolution depth given to the GLU. All convolutions have stride of 1 and paddings to maintain the same sequence length

.

8.2.1. Training details

All DeepSpeech models were trained with 0.001 learning rate using Adam optimizer until training loss converges. All mini-GCNN models were trained with 0.0003 learning rate using Adam optimizer until training loss converges. For all models, the parameters for the Adam optimizer are: $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 1e - 8$.

8.3. Data augmentation models

Signal distortion: Jimerson et al. [8] explored data augmentation via distortion of the speech signal, in which they added to the training corpus copies of the existing audio data that were modified to adjust F0 and speaking rate or to include background noise. Here we focus on modifying pitch and speaking rate using PSOLA. For pitch augmentation, the F0 of the speech signal was varied in fractions of octaves ranging from 0.10 to 0.30 with a step size of 0.05. Speaking rate was adjusted by re-sampling the audio at multiples of the sampling frequency of the utterance ranging from 0.75 to 1.25 with a step size of 0.05. Each utterance in the training corpus was distorted 10 times with parameters randomly chosen and

added to the existing training corpus, resulting in an additional 6000 minutes of audio data.

Voice conversion: The StarGAN-VC model [29] modifies the image-based StarGAN [28] to acoustic features to perform many-to-many voice conversions. StarGAN [28] implements a cycleGAN [26] architecture with an additional domain classifier, where the speaker identity was used as the domain. The VAWGAN model [30] is a two-stage framework developed for non-parallel corpus voice conversion. The first stage converts input audio to a phonetic content vector using a speaker-independent encoder. The second stage then converts the phonetic content vector to audio with the other speaker's characteristics by combining the phonetic content vector with a speaker representation input. To improve the output quality, a Wasserstein GAN was added after the decoder stage.

For each of the voice conversion methods, we selected the three speakers with the largest volume of labelled data: Speaker A (94 minutes), Speaker B (250 minutes), and Speaker C (156 minutes). Since StarGAN-VC enables many-to-many voice conversion, only one StarGAN-VC model was trained to perform voice conversion among the three speakers. The StarGAN-VC model was trained for a total of 500,000 iterations, with sample outputs taken at every 50,000 iterations to subjectively determine whether the model produced intelligible utterances. A total of six VAWGAN models were trained to perform voice conversion among the three speakers since VAWGAN only allows for one-to-one voice conversion. Each VAWGAN model was trained for 100 epochs, with samples taken every 10 epochs to determine whether synthesized utterances were intelligible. The trained StarGAN-VC and VAWGAN models were then used to convert utterances from each of the three speakers to the other two. From the original 500 minutes of audio produced by Speakers A, B, and C, we obtained 1000 minutes of synthetic data for each voice conversion model.

Figure 3 shows the mel-spectrogram of an unaltered Seneca utterance along with its corresponding synthetically generated spectrograms for randomly chosen speed and pitch distortions, as well as the StarGAN-VC and VAWGAN augmentation methods. The two voice conversion methods synthesize the utterance from speaker A as if it were spoken by speaker B. The peak locations in the two voice conversion methods are maintained, but the signature characteristics of the voice are transformed.

8.4. Language modeling

To each of our acoustic models, we apply two different language model configurations: no language model and a word-level trigram language model. The n-gram language model is trained on the transcripts of the 600 minutes of acoustic training (roughly 35,000 words) as well as an addition 6500 words of Seneca text created for language learning purposes,

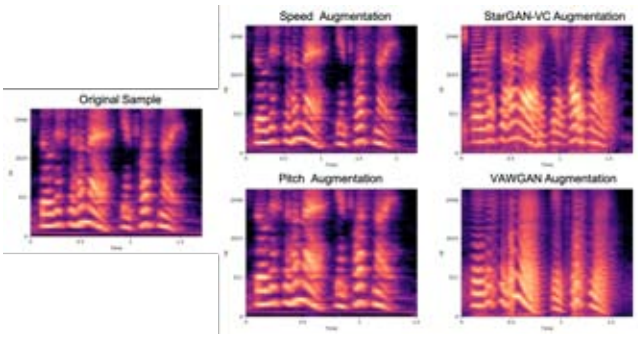

Fig. 3. Mel-spectrograms of a randomly selected utterance. The original utterance (mean F0=153, duration=2048msec) is on the left, while the various augmentation methods, counter-clockwise from top left are: pitch modification (mean F0=179), speaking rate modification (duration=3132msec), VAWGAN, and StarGAN-VC.

produced originally as written text, or transcribed by previous researchers from recordings that are no longer available. The n-gram model was created with KenLM [32] using modified Kneser-Ney smoothing, no pruning, and an n-gram order of 3.

8.5. Discussion of Results

We performed experiments on both recurrent and convolution acoustic models and found that convolutional methods are not only more compute-efficient but also yield lower word and character error rates. Unsurprisingly, we found that using n-gram language models during decoding results in large gains in accuracy, regardless of the acoustic model used. Our augmentation model using StarGAN-VC with recurrent acoustic models yields results approaching traditional GMM/HMM methods, which have generally outperformed deep learning approaches when training resources are very limited. Perhaps more significant is our finding that convolutional acoustic modeling approaches perform better than traditional approaches regardless of the augmentation technique used. In our future work, we plan to explore the utility of the mini-GCNN framework in other low-resource scenarios in order to test the limits of data scarcity and to determine whether the utility of this approach is dependent on specific typological characteristics.

Our three data augmentation methods yield varying results. This suggests that there is potential utility in combining all three augmentation models, either sequentially or collectively, an approach we will explore for future work. We also plan to build Seneca speech synthesis models in order to generate new Seneca audio from the Seneca text data for which there are no corresponding recordings. Finally, we hope to be able to use transfer learning to leverage a much larger corpus of recordings of Mohawk, a language very closely related to Seneca that is more widely spoken.

Unlike many languages that are considered "low-resource", such as Vietnamese or Haitian Creole, Seneca has essentially no written text for training language models beyond what we already included in our models. In addition, Seneca has an unusually complex morphology, which makes it very difficult to provide good vocabulary coverage, resulting in very high out-of-vocabulary rates. To address these challenges, we are investigating the feasibility of data augmentation in the language model. Our current work focuses on generating synthetic text data from both deep and n-gram character, word, and morph language models, and determining the best way to incorporate these language models into decoding. We anticipate our research will enable low and high-resource languages alike, to take advantage of the recent ASR gains afforded by deep methods.