Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020), pages 126–130
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Fully Convolutional ASR for Less-Resourced Endangered Languages

Bao Thai[†]**, Robbie Jimerson**[†]**, Ray Ptucha**[†]**, Emily Prud'hommeaux**[†‡]
[†]Rochester Institute of Technology, Rochester NY, USA
[‡]Boston College, Chestnut Hill MA, USA
{btt4530,rcj2772,rwpeec,emilypx}@rit.edu

Abstract

The application of deep learning to automatic speech recognition (ASR) has yielded dramatic accuracy increases for languages with abundant training data, but languages with limited training resources have yet to see accuracy improvements on this scale. In this paper, we compare a fully convolutional approach for acoustic modelling in ASR with a variety of established acoustic modeling approaches. We evaluate our method on Seneca, a low-resource endangered language spoken in North America. Our method yields word error rates up to 40% lower than those reported using both standard GMM-HMM approaches and established deep neural methods, with a substantial reduction in training time. These results show particular promise for languages like Seneca that are both endangered and lack extensive documentation.

Keywords: automatic speech recognition, endangered languages, indigenous languages

1. Introduction

Improvements and breakthroughs in deep learning for automatic speech recognition (ASR) have resulted in significant improvements in ASR performance in high-resource languages such as English and Mandarin (Hinton et al., 2012; Hannun et al., 2014; Chan et al., 2016; Audhkhasi et al., 2018; Chiu et al., 2018). Such methods, however, require very large volumes of labelled training data to achieve these notable results. Most languages of the world, even those with tens of millions of speakers, do not have the quantities of data required to train such systems. The data sparsity problem is even more dire for the many indigenous languages that have historically been undocumented for political or cultural reasons. Deep learning ASR systems for languages with truly limited labelled training data typically 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) frameworks.

While convolutional neural networks (CNNs) have demonstrated superior performance on vision tasks such as image classification, image segmentation, and object recognition, deep learning for ASR has relied heavily upon variants of recurrent neural networks (RNNs). In RNNs, information from timesteps before, and after in the case of bidirectional networks, is used in making the decision of the current timestep. CNNs are excellent at extracting regional patterns but typically require inputs to be of fixed size. However, as seen in object detection and image segmentation applications, fully convolutional variations can operate on multiple locations simultaneously and allow variable-size inputs.

In this paper, we present a convolutional acoustic model for ASR in low-resource conditions. We demonstrate our approach using a corpus of 10 hours of recordings of the Seneca language, a critically endangered, morphologically complex language spoken in the northeastern part of North America. Our model reduces the computational cost in terms of number of parameters while still capturing enough temporal dependencies to make accurate predictions. We find that our fully convolutional acoustic model yields significant accuracy improvements over both deep recurrent and HMM/GMM models. To demonstrate the robustness of our approach, we additionally apply our framework to Iban, an unrelated low-resource language with a phonetic inventory roughly the size of Seneca's but with a less complex morphology.

Our main contributions are as follows: 1) We introduce a deep convolutional architecture optimized for low-resource scenarios that captures feature-rich audio data over a broad temporal receptive field; 2) We utilize a fully convolutional framework for arbitrary length sequence processing; and 3) We show the effectiveness of utilizing transfer learning and data augmentation for further reducing word and character error rates.

2. Previous Work

When given sufficient in-domain monolingual training data, deep neural network methods for ASR often perform significantly better than traditional methods based on HMMs and GMMs (Hinton et al., 2012; Graves et al., 2013; Hannun et al., 2014; Amodei et al., 2016; Chan et al., 2016; Zhang et al., 2017; Chiu et al., 2018; Agenbag and Niesler, 2019). Common approaches for deep learning ASR rely on RNNs: sequence-to-sequence models like that in Chan et al. (2016) use RNNs to generate a latent representation of the utterance before decoding with RNNs, while DeepSpeech 1 and DeepSpeech 2 (Hannun et al., 2014; Amodei et al., 2016) use RNNs to capture temporal dependencies before making predictions for each timestep. Methods that produce characters, such as versions of DeepSpeech, currently use Connectionist Temporal Classification (CTC) (Graves et al., 2006) to reduce streams of characters to plausible words by combining consecutive similar characters and pauses during speech.

Convolutional architectures have achieved remarkable results in computer vision tasks such as image classification (Szegedy et al., 2015; Xie et al., 2017). Szegedy et al. (Szegedy et al., 2015) introduced the concept of an Inception block which consists of multiple filter sizes in a layer to capture different levels of regional dependencies. This concept can be applied to sequential data like speech by using filters with different widths to simultaneously capture different temporal dependencies. The Inception network introduces $1\times$ bottleneck filters to reduce the number of parameters in a model. Xie et al. (Xie et al., 2017) use Inception-like blocks but with similar filter sizes while adding skip connections similar to ResNet to allow for better gradient flow.

Previous experiments have shown that transfer learning from a model trained on resource-rich languages can improve the performance of ASR for low-resource languages (Gales et al., 2014; Imseng et al., 2014). Using synthetic data has also been found to yield improvements in true low-resource, artificially low-resource, and resource-rich conditions (Tüske et al., 2014; Billa, 2018; Wiesner et al., 2018). Carmantini et al. (Carmantini et al., 2019) introduced sample overgeneration during initialization for low-resource ASR for improved semi-supervised training on lattice-free maximum mutual information (LF-MMI) (Manohar et al., 2018). Malhotra el at. (Malhotra et al., 2019) selected samples with lower confidence in an active learning scenario for low-resource ASR.

Rosenberg et al. (Rosenberg et al., 2017) investigated the use of a CTC-based RNN and an RNN Encoder-Decoder network in character-based end-to-end ASR for low-resource languages. While recurrent-based models have demonstrated usefulness in ASR and other sequence modeling tasks, these models cannot easily take advantage of parallelization on modern hardware since the output of an RNN cell at each timestep depends on the results from the previous timestep. To mitigate this problem, Collobert et al. (Collobert et al., 2016) relies on convolution to capture temporal dependencies.

The fully convolutional, character-based architecture proposed by Collobert et al. (Collobert et al., 2016) still requires training models with large numbers of parameters. Additionally, these models have a high number of layers causing the models to converge more slowly. Our proposed model aims to reduce the complexity of the model without reducing performance by using bottleneck filters and skip connections. Additionally, instead of relying on different layers to capture different levels of temporal dependencies, we combine filters with different widths into one layer to reduce the number of layers in the model while still maintaining a wide context window. While transfer learning and data augmentation separately have both shown improvements, we explore the effectiveness of combining both concepts on low resource ASR, as well as a final fine-tuning step using only unaugmented data to prevent digital artifacts in augmented data from degrading performance.

3. Data

We conduct our experiments on Seneca, a morphologically complex and critically endangered language spoken by indigenous people in what is now Western New York State and Ontario. Although the language was still widely spoken in the Seneca community as recently as 75 years ago, Seneca children in the mid-twentieth century were typically required to attend state-run residential schools where they were punished or beaten for using their native language. Today, roughly 50 elderly individuals speak Seneca as their first language, and a few hundred others are second language speakers. There are several ongoing efforts to revitalize the Seneca language, including language immersion programs for adults and children, but there are very few available Seneca recordings or texts, as many members of the Seneca community are reluctant to allow their speech to be recorded or transcribed. One motivation for developing a robust ASR system for Seneca is to accelerate efforts to document the language while there are living native speakers and to produce educational materials for the immersion programs that will train the next generation of speakers.

The available transcribed audio recordings consist of approximately 720 minutes of spontaneous, naturalistic speech produced by eleven adult speakers, eight male and three female. All speakers in the dataset are middle-aged or elderly first-language Seneca speakers whose second language is English. Recordings were made over many years primarily by Seneca language learners under a variety of conditions using various recording equipment, resulting in a diverse range of audio quality.

The recordings were transcribed using Seneca's current orthography, which uses 30 characters, and segmented at the utterance level by second-language Seneca speakers. Since Seneca orthography is quite reliably phonemic, with few ambiguous character-to-phone and phone-to-character mappings, we choose to treat characters (excluding punctuation) as phones. Using utterance boundaries provided in the reference transcripts, we randomly selected individual utterances from the full corpus of 720 minutes until we had obtained 600 minutes of audio for training. The remaining 120 minutes made up the test set. We deliberately selected utterances at random to maximize diversity in terms of gender, age, dialect, voice quality, and content (e.g. narrative vs. conversation) of both the train and test sets in order to avoid overfitting to any particular speaker or speaker characteristics. While this selection procedure lead to certain speakers appearing in both the testing and training sets, we were obliged to make this compromise due to the limited number of speakers of the language. In addition to the transcriptions of the recorded audio (roughly 35,000 words), we have available text data consisting of 6000 words of previously transcribed texts for which no corresponding audio is available.

To demonstrate the generalizability of our methods, we also conduct our experiments on Iban, a Malayic language spoken in Brunei and Malaysia. The publicly available dataset ((Juan et al., 2014)) consists of 479 minutes of professional recordings of broadcast news, partitioned into 408 minutes of training data and 71 minutes of testing data. There are 17 speakers (7 male, 10 female) in the training set and 6 speakers (2 male, 4 female) in the test set.

4. Methods

4.1. Acoustic Modeling

We utilize a fully convolutional acoustic model constructed from a family of one dimensional convolution layers. The model takes either 13 MFCCs and their first and second derivatives, or 80 log mel-filterbanks as input features. Both are obtained using 25ms windows with 10ms stride.

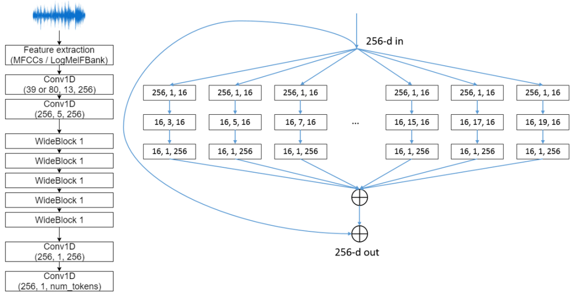

Figure 1: **Left:** The overall architecture of our convolutional approach. **Right:** A WideBlock consisting of 9 paths, each consisting of bottleneck filters centered by filters of different width to capture different levels of temporal dependencies. Each layer is shown as (# input channels, filter width, # output channels).

Figure 1 shows the overall network architecture and the architecture of a WideBlock, the main building block of our model. The details of each are described next.

WideBlock: The main building block of our architecture is the WideBlock (Figure 1), named for the high number of paths in each block. The architecture of the block, taking inspiration from ResNeXt blocks used in image classification (Xie et al., 2017), consists of several parallel streams, each consisting of bottleneck 1×1 convolution layers before and after a normal convolution layer. The bottleneck layers reduce the complexity of the model by reducing the number of parameters required by the middle convolution operation. Instead of keeping the same filter size for all paths, we draw inspiration from Inception networks and employ filters with different sizes in each layer. The filter widths are odd numbers between 3 and 19. This choice is suitable for speech-related tasks since temporal dependencies in audio typically have more variance than spatial dependencies in visual tasks. The different filter sizes allow the model to pick up both short-term and long-term temporal dependencies. The output from each path is then summed before being added to the input of each block, forming a skip connection.

Acoustic Model: Our acoustic model consists of two convolutional layers between the input feature vector and the first WideBlock (Figure 1). These embedding layers convert input audio features into a vector of desired depth and temporal content. The acoustic architecture continues with five WideBlocks, then two 1×1 convolution layers which act as fully-connected layers. The final layer outputs a vector with size corresponding to the number of tokens to be predicted. Batch normalization and ReLU are used after each convolution operation. To prevent overfitting due to limited data, dropout layers of 0.25 are added after each WideBlock. To train the network, the CTC loss function is used.

DeepSpeech: To compare the performance of our deep approach against recurrent-based ASR models, we also trained a DeepSpeech model. The DeepSpeech model consists of a five-layer recurrent neural network with Long-Short Term Memory 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. The DeepSpeech model uses the same input features as our deep approach and also uses CTC loss.

Kaldi: We also compare the performance of our model against the traditional HMM/GMM framework provided by Kaldi (Povey et al., 2011) with a triphone acoustic model trained with the parameter settings described in the Kaldi tutorial and a word-level trigram language model. A second acoustic model was created using Kaldi's time-delay neural network (TDNN) architecture trained with the lattice-free maximum mutual information (LF-MMI) objective function (Peddinti et al., 2015).

4.2. Multistage Learning

Transfer learning has proven successful in deep learning tasks with limited domain data. We extend this concept with a multistage transfer learning strategy. In the first stage, we train an acoustic model on a 960-hour LibriSpeech English corpus for 100 epochs. In the second stage, weight initialization is from the model obtained in the first stage. The model was then trained on heavily augmented training data as per (Jimerson et al., 2018) for 100 epochs or until convergence. In the final stage, the weights of the model from the second stage were used to initialize a model which is trained only on unaugmented data. For this final stage, the learning rate is reduced by an order of magnitude.

5. Results

Table 1 shows the performance for Seneca across different acoustic models with different transfer learning and augmentation strategies. To evaluate the performance of each model, we use word error rate (WER) and character error rate (CER). WER is the minimum edit distance over a word alignment, aggregated across utterances and normalized by the total number of words in the reference transcript. CER is calculated by aggregating the character-level minimum edit distance over all utterances and normalizing by the total number of characters in the reference. We report results for decoding both with and without a trigram language model built on the transcripts of the 10 hours of acoustic training data using KenLM (Heafield, 2011) with modified Kneser-Ney smoothing and no pruning.

Table 1 shows that DeepSpeech (DS) with no transfer learning, augmentation, or language model yields little or no correct output. With a language model, the WER and CER for this model are reduced, but results are still mostly incorrect. Our deep approach shows slightly better performance than DeepSpeech without a language model and significantly lower WER when decoding with a trigram language model.

	DS (NO LM)		DS (W/LM)		Our CNN (NO LM)		Our CNN (W/LM)	
	WER	CER	WER	CER	WER	CER	WER	CER
No TL, no Aug (Baseline)	1.000	0.891	0.970	0.872	0.839	0.365	0.421	0.257
TL, no Aug (+TL)	0.859	0.436	0.727	0.409	0.785	0.328	0.337	0.199
TL + Aug (+TL,Aug)	1.000	0.716	0.975	0.698	0.730	0.303	0.319	0.194
TL + Aug + finetune (+TL,Aug,FT)	0.850	0.427	0.693	0.421	0.699	0.278	**0.299**	**0.175**

Table 1: Seneca WER and CER for various transfer learning (TL), augmentation (Aug), and fine-tuning (FT) strategies (rows) vs. DeepSpeech (DS) and our deep CNN architecture without (NO LM) and with (W/LM) a trigram language model.

	NO LM		W/LM	
	WER	CER	WER	CER
Baseline	0.784	0.307	0.369	0.197
+TL	0.768	0.309	0.302	0.172
+TL,Aug	0.758	0.324	0.307	0.187
+TL,Aug,FT	0.656	0.247	**0.243**	**0.130**

Table 2: Seneca WER and CER using our deep CNN approach with log mel-filterbank feature as input features with and without a trigram language model.

Acoustic Model	WER
Monophone GMM/HMM	0.608
Triphone GMM/HMM	0.524
TDNN LF-MMI	0.421

Table 3: Seneca WER for Kaldi HMM-GMM models and TDNN with LF-MMI.

	NO LM		W/LM	
	WER	CER	WER	CER
Baseline	0.856	0.463	0.487	0.286
+TL	0.668	0.287	0.413	0.257
+TL,Aug	0.665	0.226	0.420	0.286
+TL,Aug,FT	0.518	0.160	**0.266**	**0.116**

Table 4: Iban WER and CER for transfer learning and augmentation strategies within our architecture using with log mel-filterbanks as input features with and without trigram language model built using only the transcripts of the audio.

Acoustic Model	WER
Monophone GMM/HMM	0.372
Triphone GMM/HMM	0.265
TDNN LF-MMI	0.175

Table 5: Previously reported WER for Iban 2 HMM-GMM models and TDNN with LF-MMI, all decoded with a language model built on the full 2-million word text corpus.

Using transfer learning from a high resource language improves performance across all models and all language model settings. Training on augmented data after transfer learning from a high resource language degrades the performance of DeepSpeech models in terms of WER but improves CER. For our deep architecture, this configuration improves results across the board. In all configurations for Seneca, our deep approach substantially outperforms the corresponding DeepSpeech model.

Fine-tuning of the augmented model using only non-augmented data yields the best performance across all models, with a WER of 0.299 using our deep acoustic model. While fine-tuning after augmentation results in improvements, it yields much larger absolute and relative reductions in WER for the DeepSpeech model than for our deep architecture. Table 2 shows results of using log mel-filterbank features in place of MFCCs with modest improvement.

Table 3 shows three Kaldi results on this same dataset: two standard HMM/GMM models (monophone and triphone) and one deep architecture, TDNN with LF-MMI. For Seneca, our deep architecture substantially outperforms all three of these models, including the TDNN.

Demonstrating the efficacy and generalizability of our models on other low-resource datasets, Table 4 shows the performance of our deep method under different configurations for the Iban language. We see slightly higher but comparable error rates on this dataset, which had three fewer hours of acoustic training data.

Table 5 shows previously reported results [1] for the three Kaldi models for Iban. These results are noticeably lower than those we report using the same acoustic model training configurations for Seneca. In addition, the TDNN LF-MMMI model yields a lower error rate than our best deep model. We note that the language model used to decode with these Kaldi models was built on a 2-million word text corpus, while the results presented above in Table 4 for our own deep methods used a language model built using only the transcripts from the 7 hours of available audio data. We suspect that this accounts for much of this discrepancy. It is also possible that our framework is better suited to the lower-quality recordings typical in the Seneca dataset and less appropriate for the clean, professionally recorded Iban data. We also note that our model yields comparable WER error rates in both languages, which points to its superior ability to generalize to new datasets.

6. Conclusions

In this paper, we introduced a residual network with a very wide filter selection in a fully convolutional architecture for low-resource ASR acoustic modeling. We show that our acoustic model outperforms a typical recurrent-based deep neural network in all experimental settings while also being more compute-efficient. Our deep acoustic model, when combined with a trigram language model, outperforms the

[1] https://github.com/bagustris/id

traditional GMM/HMM model without the need for transfer learning or data augmentation. We also show that transfer learning from a high-resource language and data augmentation contribute to meaningful reductions in word error rate achieved by the model for two distinct low-resource languages. Our results point the way toward new, fast-training deep learning ASR methods for languages with extremely limited audio and textual training resources.

7. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1761562. We recognize the contributions of the team collecting and analyzing this dataset, including Morris Cooke, Richard Hatcher, Alex Jimerson, Mike Jones, Megan Kennedy, Whitney Nephew, Aryien Stevens, and Karin Michelson. We are grateful for the cooperation, support, generosity of the elders of the Seneca Nation of Indians.

8. Bibliographic References

Agenbag, W. and Niesler, T. (2019). Automatic sub-word unit discovery and pronunciation lexicon induction for asr with application to under-resourced languages. *Computer Speech & Language*, 57:20–40.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., and Chen, G. (2016). Deep Speech 2: End-to-end speech recognition in English and Mandarin. In *ICML*, pages 173–182.

Audhkhasi, K., Kingsbury, B., Ramabhadran, B., Saon, G., and Picheny, M. (2018). Building competitive direct acoustics-to-word models for english conversational speech recognition. In *ICASSP*, pages 4759–4763.

Billa, J. (2018). Isi asr system for the low resource speech recognition challenge for indian languages. *Interspeech*, pages 3207–3211.

Carmantini, A., Bell, P., and Renals, S. (2019). Untranscribed web audio for low resource speech recognition. *Interspeech*, pages 226–230.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*.

Chiu, C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R., Rao, K., Gonina, E., et al. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In *ICASSP*.

Collobert, R., Puhrsch, C., and Synnaeve, G. (2016). Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*.

Gales, M., Knill, K., Ragni, A., and Rath, S. (2014). Speech recognition and keyword spotting for low-resource languages: Babel project research at CUED. In *SLTU*, pages 16–23.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376.

Graves, A., Mohamed, A., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *ICASSP*, pages 6645–6649.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., et al. (2014). Deep Speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *SMT Workshop*, pages 187–197.

Hinton, G., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Kingsbury, B., and Sainath, T. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29:82–97, November.

Imseng, D., Motlicek, P., Bourlard, H., and Garner, P. (2014). Using out-of-language data to improve an under-resourced speech recognizer. *Speech Communication*, 56:142–151.

Jimerson, R., Simha, K., Ptucha, R., and Prud'hommeaux, E. (2018). Improving ASR output for endangered language documentation. In *SLTU*, pages 182–186.

Juan, S., Besacier, L., and Rossato, S. (2014). Semi-supervised g2p bootstrapping and its application to asr for a very under-resourced language: Iban. In *SLTU*, May.

Malhotra, K., Bansal, S., and Ganapathy, S. (2019). Active learning methods for low resource end-to-end speech recognition. *Interspeech*, pages 2215–2219.

Manohar, V., Hadian, H., Povey, D., and Khudanpur, S. (2018). Semi-supervised training of acoustic models using lattice-free mmi. In *ICASSP*, pages 4844–4848.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Interspeech*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi speech recognition toolkit. In *ASRU*.

Rosenberg, A., Audhkhasi, K., Sethy, A., Ramabhadran, B., and Picheny, M. (2017). End-to-end speech recognition and keyword search on low-resource languages. In *ICASSP*, pages 5280–5284.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *CVPR*, pages 1–9.

Tüske, Z., Golik, P., Nolden, D., Schlüter, R., and Ney, H. (2014). Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages. In *Interspeech*.

Wiesner, M., Renduchintala, A., Watanabe, S., Liu, C., Dehak, N., and Khudanpur, S. (2018). Low resource multimodal data augmentation for end-to-end ASR. In *Interspeech*.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500.

Zhang, Y., Chan, W., and Jaitly, N. (2017). Very deep convolutional networks for end-to-end speech recognition. In *ICASSP*, pages 4845–4849.