To appear in: Proc. 27th International Conference on Artificial Neural Networks "ICANN 2018"

Combining Articulatory Features with End-to-end Learning in Speech Recognition

Leyuan Qu, Cornelius Weber, Egor Lakomkin, Johannes Twiefel, Stefan Wermter

University of Hamburg, Department of Informatics,
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
{qu,weber,lakomkin,twiefel,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM

Abstract. End-to-end neural networks have shown promising results on large vocabulary continuous speech recognition (LVCSR) systems. However, it is challenging to integrate domain knowledge into such systems. Specifically, articulatory features (AFs) which are inspired by the human speech production mechanism can help in speech recognition. This paper presents two approaches to incorporate domain knowledge into end-to-end training: (a) fine-tuning networks which reuse hidden layer representations of AF extractors as input for ASR tasks; (b) progressive networks which combine articulatory knowledge by lateral connections from AF extractors. We evaluate the proposed approaches on the speech Wall Street Journal corpus and test on the eval92 standard evaluation dataset. Results show that both fine-tuning and progressive networks can integrate articulatory information into end-to-end learning and outperform previous systems.

Keywords: Articulatory Features, Automatic Speech Recognition, Deep Neural Networks (DNN), End-to-end Learning.

1 Introduction

End-to-end learning has been successfully applied in many domains, such as handwriting recognition [1], neural machine translation [2], and so on. Furthermore, end-to-end models have become popular in automatic speech recognition (ASR) tasks. The conventional ASR pipeline consists of many different components: the acoustic model, pronunciation model and language model. These components are separate and require lots of human expertise, e.g. a handcrafted pronunciation dictionary and designed senone states for Hidden Markov Models (HMMs). Additionally, the training targets and alignment information needed for neural networks in a DNN-HMM paradigm can only be obtained from another GMM-HMMs (GMM is short for Gaussian Mixture Model) model which is trained beforehand. Such a pipeline requires not only multiple training stages but also different optimization functions [4].

To simplify this complex paradigm, end-to-end learning approaches [4-6, 11-13] have been proposed to replace hand-designed feature engineering and jointly learn all

components in a single architecture. These approaches can be transformed into computational flow graphs which can be optimized by backpropagation in a simple end-to-end training process. End-to-end models are able to naturally handle sequences of arbitrary lengths and directly optimize the word error rate. However, it is challenging to integrate domain knowledge into these models. Therefore, the goal of this study is to combine articulatory features into end-to-end learning.

Articulatory features (AFs), also known as phonological features, phonological attributes or distinctive phonetic features, are used to represent the movement of different articulators, such as lips and tongue, during speech production. AFs can be robustly estimated from speech by statistical classifiers, such as GMM and neural networks [7]. A series of studies have demonstrated that AFs can improve the performance of ASR systems by systematically accounting for coarticulation, speaking styles and other variability, especially in a noisy scenario [8]. Conventional methods to extract AFs from speech require precise boundary transcription. To get this boundary information, the usual practice is using forced alignments generated by a GMM-HMMs model [9], or labeling data manually at a frame-level [10], which are complex and time-consuming.

Our hypothesis in this paper is that AFs can provide useful and complementary representations that cannot be learned automatically by an end-to-end architecture. This paper explores two approaches to integrate domain knowledge to improve end-to-end model performance. Our contribution is two-fold: In the first step, we train a bank of AF extractors using Connectionist Temporal Classification (CTC) in an end-to-end way, which does not require precise phone or frame-level boundary information; In the second step, we propose two approaches (fine-tuning networks and progressive networks) to integrate domain knowledge (articulatory features) into end-to-end learning in speech recognition tasks.

2 Related work

2.1 End-to-end learning in speech recognition

At present, end-to-end learning in ASR can be mainly divided into two parts: CTC-based approaches and encoder-decoder models. For the CTC, Graves et al. [5] introduced the CTC loss function which removes the alignment constraint by introducing a "blank" label and allows to train a sequence labeling task directly without alignment and pre-segmentation. Miao et al. [4] explored a weighted finite-state transducers-decoding method to incorporate lexicons and language models in CTC objective function-based models. Recently, Zweig et al. [6] presented an iterated CTC approach on the NIST 2000 conversational telephone speech evaluation set which significantly improved performance over previous systems. For the encoder-decoder, Chorowski et al. [11] introduced an attention mechanism into speech recognition, in which the authors combined both content and localization information to recognize a longer utterance. Bahdanau et al. [12] replaced HMMs with an attention-based recurrent sequence generator (ARSG) on the LVCSR task. Unlike CTC-based methods, the ARSG system can learn a language model implicitly. Chan et al. [13] presented a Listen, Attend and Spell system to transcribe speech to characters directly. They reported 10.3 % word error rate

(WER) with rescoring compared to the-state-of-the-art WER of 8.0% achieved by a convolutional neural network and long short-term memory DNN-HMMs model [20] on 2000 hours Google voice search dataset.

2.2 Domain knowledge integration in speech recognition

There are lots of approaches focusing on integrating domain knowledge to improve ASR performance, such as in feature engineering: mel-frequency cepstral coefficients [25] and vocal tract length normalization [26], and in algorithm optimization: sequence-discriminative training [27]. Here, we only consider studies that involve linguistic and phonetic knowledge.

Lee et al. [14] proposed automatic speech attribute transcription (ASAT) which is a new detected-based speech recognition paradigm. Compared to conventional ASR top-down paradigms, ASAT is bottom-up and coincident with the mechanism of humans perceiving and producing speech. To further improve phonological feature detection accuracy, Yu et al. [9] replaced one hidden layer multi-layer perceptrons by DNNs when building attribute detectors. Based on the high attribute detection precision, excellent phoneme estimate accuracy was obtained on the WSJ0 benchmark. Siniscalchi et al. [15] integrated acoustic-phonetic information into lattice rescoring. Inspired by shared phonetic knowledge among different languages, Siniscalchi et al. [16] designed a universal set of phones and used the set to improve the performance of cross-language phone recognition. Pitch accent was proposed by Ananthakrishnan et al. [17] to re-score the N-best results outputted from a standard ASR system. At present, the works integrating knowledge into ASR are mostly based on HMM hybrid architectures. Our approaches mainly focus on combining domain knowledge with neural end-to-end ASR systems.

3 Model Architecture

In this section, we present the details of AF extractors, fine-tuning networks and progressive networks.

3.1 AF extractor

Fig. 1 shows the flow diagram to get AF-level transcriptions. First, we split words into phonemes according to the CMU dict[1]. Then, we generate AFs transcriptions according to the mapping [9] (see Table 3 in the Appendix). The AF-level transcriptions will be used as training targets to build the AF extractors.

Eight AF extractors were built: place, manner, anterior, back, continuant, round, tense and voiced. The AF extractor architecture is shown in Fig. 2 (a), which begins with two layers of 2D convolutions, followed by five layers of gated recurrent units (GRU), and the output layer is a fully connected layer. We train each extractor with the

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

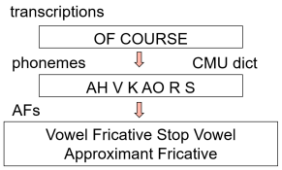

Fig. 1. Flowchart to convert word-level transcriptions of the phrase “of course” to AF labels.

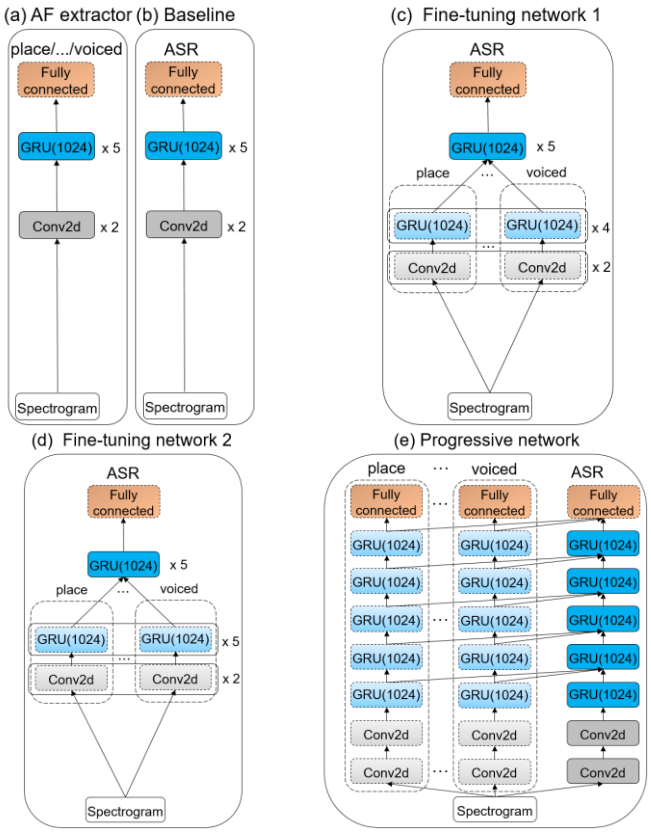

Fig. 2. Illustration of (a) AF extractor, (b) ASR baseline system, (c) and (d) fine-tuning networks and (e) progressive networks. The ASR baseline system is based on Deep Speech 2 [19]. Note: frozen (dotted line) without backpropagation and weight updating.

CTC and additional two symbols (blank and space). For example, for ‘voiced’, the target labels are {voiced, other, space, blank}.

3.2 Fine-tuning Networks

Fine-tuning is a process to transfer what a neural network learned on a given task to a second task. In this paper, AF extractors that have been learnt in a first task can be treated as a fixed front-end which transforms spectrograms to AFs. Hidden layer outputs from different AF extractors will be combined, then fed into another neural network for the second task (ASR). Fig. 2 (c) and (d) show the fine-tuning networks used in this study. The details of AF extractors (place, manner, anterior, back, continuant, round, tense and voiced) are shown in Fig. 2 (a). We concatenate the fourth or fifth GRU layer output of all extractors as a vector, namely fine-tuning networks 1 (Fig. 2 (c)) and fine-tuning networks 2 (Fig. 2 (d)) respectively, and feed it into a 5 bidirectional GRU-layer neural network for the ASR task.

3.3 Progressive Networks

Progressive networks with lateral connections from previous tasks can accelerate learning speed and avoid forgetting [18]. They not only learn relevant features but also acquire different representations from previous learned tasks, which may be irrelevant to the target task. The scheme of progressive networks is shown in Fig. 2 (e). In this paper, there are no connections between the AF extractors and they are trained in parallel and independently, then linearly combined. The source task is AF extraction from speech signals and the target task is speech recognition. We use the following formula to compute outputs of layer i in ASR tasks:

$$h_i = W_i(h_{i-1} + \sum_{j=1}^{8} k_{i-1}^{j}) \tag{1}$$

where h_i is the output of layer i of the ASR system, k_i^j is the output of layer i of AF extractor j, $W_i \in R^{n_i \times n_{i-1}}$ is the weight matrix of layer i of ASR systems, with n_i the number of units at layer i. Layer h_i receives input from both h_{i-1} and k_{i-1}^j via equation (1).

4 Experiments

In this section, we present the dataset and the experimental setup.

4.1 Evaluation Metric

In this paper, we use the word error rate (WER) to evaluate model performance. WER quantifies how many elementary operations are required to transform the generated output sequence of the network into the correct target sequence. It is calculated as follows:

$$WER = \frac{S + D + I}{N} \tag{2}$$

where S is the number of substitutions, D is the number of deletions and I is the number of insertions. N is the total number of words in the reference.

4.2 Data

We used the Wall Street Journal (WSJ) [22] speech corpus both for AF and ASR experiments. The training set is the 81 hours 'train-si284' with about 37K sentences. We used the 'dev93' development set for validation and hyper-parameter optimization and report the final performance on the 'eval92' test set.

4.3 Training

The baseline ASR system (shown in Fig. 2 (b)) used in this paper is similar to the Deep Speech 2 system [19]. The first two layers of all architectures are 2D (frequency and time domains) convolutions. The convolution layers not only reduce temporal variability in the time domain but also normalize speaker variance in the frequency domain [23]. These are followed by GRU layers. It has been shown that GRU cells achieve comparable performance to Long Short-Term Memory (LSTM) but GRU cells are faster and easier to train [21]. Finally, we pass the output from the GRU cells to a fully-connected layer.

The input features for all models are spectrograms derived from the raw audio files, with 20ms window size and 10ms window stride. All neural networks are trained with the CTC, using the stochastic gradient descent optimization strategy along with a mini-batches of 20 utterances per batch. We use 40 epochs and pick the model that performs best on the development set to evaluate on the test set. Learning rates are chosen from [1e-4, 6e-4], and a learning rate annealing algorithm is used by the value of 1.1 after each epoch. The momentum is 0.9. Batch normalization is used to optimize models and accelerate training on hidden layers. All architectures described in this paper do not use language models and add 'space' to segment outputs into words. The output alphabet for ASR experiments consists of 29 classes (a, b, c, …, z, space, apostrophe, blank). Once all AF extractors have been built, we freeze all extractor weights during ASR training. All models are trained on the corpus described in 4.1.

5 Results and Discussion

In this section, we present the performance of AF extractors and ASR systems using fine-tuning networks and progressive networks. Table 1 shows the error rate of different AF extractors trained on the 81 hours 'train-si284' training set. All error rates are less than 10%, from which we conclude that articulatory features can be robustly detected from speech signals using the CTC loss function without requiring boundary alignment information.

Table 1. Results of articulatory feature extractors at a phoneme-level.

	Articulatory Features		Error Rate (%)
Place	Vowel Fricative Nasal	Stop Approximant	9.4
Manner	Coronal High Dental Glottal Labial	Low Mid Retroflex Velar	8.6
Others	Anterior		5.2
	Back		9.2
	Continuant		4.0
	Round		9.1
	Tense		8.7
	Voiced		4.0

Table 2 lists the results from our ASR experiments and some results as reported in previous approaches using the CTC loss function on the WSJ benchmark. The fine-tuning network 1 (using 4-layer GRU from AF extractors) achieves a 33.2% WER which is worse than the baseline model (32.4%). However, when concatenating 5 layers of output from all AF extractors, the fine-tuning network 2 performs both better than the fine-tuning network 1 and the baseline system. We hypothesize that the deeper fine-tuning network 2 can capture more invariant and effective articulatory representation than the architecture with shallow layers.

The progressive network performs best in all our approaches achieving 28.6% WER. The progressive network can avoid forgetting and provide some complementary articulatory representations which can be learned by end-to-end architectures.

Table 2. Word Error Rate (WER) on the Wall Street Journal Corpus "eval92 20k" evaluation set. All models are trained with CTC loss function. No language models are used but the CTC-lexicon model [4] uses a lexicon.

Model	WER (%)
RNN-CTC [5]	30.1
BDRNN-CTC [24]	35.8
CTC-lexicon [4]	26.9
Baseline	32.4
Fine-tuning network 1	33.2
Fine-tuning network 2	31.6
Progressive network	28.6

To examine the approaches we proposed and make a fair comparison, we cite some previous approaches which use CTC and an end-to-end architecture, and only compare

the ASR performance without additional language models. Compared to prior approaches, the final performance of our progressive network (28.6%) is better than the bidirectional RNN model [19] (35.8%) and the RNN-CTC approach (30.1%). It is not as good as the CTC lexicon system [4] (26.9%) which uses a lexicon in decoding and the lexicon helps to correct the output to correctly spelled words but we do not.

6 Conclusions and Future Work

In this work, we have presented two approaches to combine domain knowledge AFs into end-to-end learning. First, fine-tuning neural networks are proposed to concatenate hidden layer outputs of AF extractors as inputs to another RNN for ASR. Second, a progressive neural network with lateral connections from AF extractors is proposed to integrate articulatory knowledge into an end-to-end architecture. Results show that both approaches can effectively incorporate articulatory information into end-to-end learning. Furthermore, the progressive neural network brings a significant improvement compared to the baseline system and to previous works.

Different speech attributes play different roles during speech production. Future work will investigate the weighted combination approach to automatically learn the contributions of different speech attributes. Furthermore, we are interested to integrate more domain knowledge into end-to-end learning under noisy and reverberation scenarios. The integration of AF improves ASR performance while increasing computation and time complexity. Future work will also focus on jointly training different AF extractors with one network to decrease computation and time complexity.

Acknowledgements

The authors gratefully acknowledge partial support from the China Scholarship Council (CSC), the German Research Foundation DFG under project CML (TRR 169), and the European Union under project SECURE (No 642667).

References

1. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324 (1998).
2. Bahdanau, D., Cho, K., and Bengio, Y.: Neural machine translation by jointly learning to align and translate. In Proceedings of the ICLR (2015).
3. Wang K., Babenko B., Belongie S.: End-to-end scene text recognition. In: Proceedings of ICCV-2011pp. 1457-1464 (2011).
4. Miao, Y., Metze, F.: End-to-End Architectures for Speech Recognition. New Era for Robust Speech Recognition. Springer, Cham, 299-323 (2017).
5. Graves A., Fernández S., Gomez F., et al.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of ICML-2006, pp. 369-376 (2006).
6. Zweig, G., Yu, C., Droppo, J., et al.: Advances in all-neural speech recognition. In: Proceedings of ICASSP-2017, pp. 4805-4809 (2017).

7. King, S., Taylor, P.: Detection of phonological features in continuous speech using neural networks. Computer Speech & Language, 14(4): 333-353 (2000).
8. Kirchhoff, K.: Robust Speech Recognition Using Articulatory Information, PhD Thesis, University of Bielefeld (1999).
9. Yu, D., Siniscalchi, S. M., Deng, L., et al.: Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition. In: Proceedings of ICASSP-2012, pp. 4169-4172 (2012).
10. Sak, H., Senior, A., Rao, K., et al.: Learning acoustic frame labeling for speech recognition with recurrent neural networks. In: Proceedings of ICASSP-2015, pp. 4280-4284 (2015).
11. Chorowski, J. K., Bahdanau, D., Serdyuk, D., et al.: Attention-based models for speech recognition. Advances in Neural Information Processing Systems. 577-585 (2015).
12. Bahdanau, D., Chorowski, J., Serdyuk, D., et al.: End-to-end attention-based large vocabulary speech recognition. In: Proceedings of ICASSP-2016, pp. 4945-4949 (2016).
13. Chan, W., Jaitly, N., Le, Q., et al.: Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In: Proceedings of ICASSP-2016, pp. 4960-4964 (2016).
14. Lee, C.-H., Clements, M. A., Dusan, S., Fosler-Lussier, E., Johnson, K., Juang, B.-H., Rabiner, L. R.: An overview on automatic speech attribute transcription (ASAT), In: Proceedings of INTERSPEECH-2007, pp. 1825-1828 (2007).
15. Siniscalchi, S. M., Lee, C.-H.: A study on integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition, Speech Communication, vol. 51, pp. 1139-1153 (2009).
16. Siniscalchi, S. M., Lyu, D. C., Svendsen, T., et al.: Experiments on cross-language attribute detection and phone recognition with minimal target-specific training data. IEEE transactions on audio, speech, and language processing, 20(3): 875-887 (2012).
17. Ananthakrishnan, S., Narayanan, S.: Improved speech recognition using acoustic and lexical correlates of pitch accent in a n-best rescoring framework. In: Proceedings of ICASSP-2007, 4: IV-873-IV-876 (2007).
18. Rusu, A. A., Rabinowitz, N. C., Desjardins, G., et al.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016).
19. Amodei, D., Ananthanarayanan, S., Anubhai, R., et al.: Deep speech 2: End-to-end speech recognition in English and Mandarin. In: Proceedings of ICML-2016, pp. 173-182 (2016).
20. Sainath, T., N., Vinyals,. O., Senior, A., et al. Convolutional, long short-term memory, fully connected deep neural networks. In: Proceedings of ICASSP-2015, pp. 4580-4584 (2015).
21. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: Proceedings of ICML-2015, pp. 2342-2350 (2015).
22. Paul, D., B., Baker, J., M.: The design for the Wall Street Journal-based CSR corpus. Proceedings of the workshop on Speech and Natural Language. pp. 357-362 (1992).
23. Abdel-Hamid, O., Mohamed, A., Jiang, H., et al.: Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In: Proceedings of ICASSP-2012, pp. 4277-4280 (2012).
24. Hannun, A., Y., Maas, A., L., Jurafsky, D., et al.: First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. arXiv preprint arXiv:1408.2873 (2014).
25. Davis, S., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In: Proceedings of ICASSP-2015, pp. 357-366 (1980).
26. Lee, L., Rose, R.: A frequency warping approach to speaker normalization. IEEE Transactions on Speech and Audio Processing, 6(1):49–60 (1998).
27. Veselý, K., Ghoshal, A., Burget, L., et al.: Sequence-discriminative training of deep neural networks. In: Proceedings of INTERSPEECH-2013, pp. 2345-2349 (2013).

Appendix

Table 3 shows the details of eight AF extractors (Manner, Place, Anterior, Back, Continuant, Round, Tense, Voiced). Output units states the number of units in each AF extractor output layer. The phoneme-level transcriptions shown in the last column can be transformed into AF-level labels according to the flow diagram shown in Fig. 1 when building AF extractors.

Table 3. The mapping of articulatory features and phonemes used in this paper [9].

AF extractor number	Output units	Category	Attribute	Phonemes
1	39	Manner	Vowel	iy ih eh ey ae aa aw ay ah ao oy ow uh uw er
			Fricative	jh ch s sh z zh f th v dh hh
			Nasal	m n ng
			Stop	b d g p t k
			Approximant	w y l r
2	41	Place	Coronal	d l n s t z
			High	ch ih iy jh sh uh uw y ow g k ng
			Dental	dh th
			Glottal	hh
			Labial	b f m p v w
			Low	aa ae aw ay oy
			Mid	ah eh ey ow
			Retroflex	er r
			Velar	g k ng
3	14	Other	Anterior	b d dh f l m n p s t th v z w
4	11		Back	ay aa ah ao aw ow oy uh uw g k
5	26		Continuant	aa ae ah ao aw ay dh eh er r ey l f ih iy oy ow s sh th uh uw v w y z
6	10		Round	aw ow uw ao uh v y oy r w
7	19		Tense	aa ae ao aw ay ey iy ow oy uw ch s sh f th p t k hh
8	29		Voiced	aa ae ah aw ay ao b d dh eh er ey g ih iy jh l m n ng ow oy r uh uw v w y z