Interspeech 2018
2-6 September 2018, Hyderabad

Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning

Abhinav Jain, Minali Upreti, Preethi Jyothi

Department of Computer Science and Engineering, Indian Institute of Technology Bombay, India
{abhinavj,idminali,pjyothi}@cse.iitb.ac.in

Abstract

One of the major remaining challenges in modern automatic speech recognition (ASR) systems for English is to be able to handle speech from users with a diverse set of accents. ASR systems that are trained on speech from multiple English accents still underperform when confronted with a new speech accent. In this work, we explore how to use accent embeddings and multi-task learning to improve speech recognition for accented speech. We propose a multi-task architecture that jointly learns an accent classifier and a multi-accent acoustic model. We also consider augmenting the speech input with accent information in the form of embeddings extracted by a separate network. These techniques together give significant relative performance improvements of 15% and 10% over a multi-accent baseline system on test sets containing seen and unseen accents, respectively.

Index Terms: Accented speech recognition, accent embeddings, multi-task learning.

1. Introduction

Accents are known to be one of the primary sources of speech variability [1]. This poses a serious technical challenge to ASR systems, despite impressive progress over the last few years. A real-world challenge that still remains for ASR systems is to be able to handle unseen accents which are absent during training. This work focuses on building solutions for handling accent variability, and in particular the challenge of unseen accents.

We expect variability due to accent to exhibit characteristics different from variability across speakers, or variability due to disfluencies and other speech artifacts. Unlike speech artifacts, an accent is typically present through out an utterance. Unlike variability across speakers, accents tend to fall into linguistic classes (correlated with speakers' native languages). Handling variability across accents is more complex than these other variabilities: Even humans typically require exposure to the same or similar accents before recognizing speech in a new accent well enough. So a natural approach to training a neural network for accented speech recognition is to expose it to different accents.

We draw a distinction between simply exposing the neural network to multiple accents, and making it *aware of* different accents. The former is achieved by simply drawing the training samples from multiple accents. The network could form a model of accents from this data. But our thesis in this work is that we can do better by actively helping the network learn about accents. We develop two complementary approaches for building accent awareness – asking the learner and telling the learner.

- *Asking the learner:* We use a *multi-task training framework* to build a network that not only performs ASR, but also predicts the accent of the utterance.
- *Telling the learner:* We use a separately trained network which extracts accent information (in the form of an embedding) from the speech, and then we make this information available to the ASR network.

We also combine both approaches by feeding the auxiliary accent embeddings as input to the multi-task network and observe additional gains on speech recognition performance.

2. Related Work

Improving recognition performance on accented speech has been explored fairly extensively in prior work. One of the earliest approaches involved augmenting a dictionary with accent-specific pronunciations learned from data, which significantly reduced cross-accent recognition error rates [2]. Multiple accents in languages other than English, such as Chinese and Afrikaans, have also been studied in prior work [3, 4, 5, 6].

For accented speech recognition, initial approaches on adapting acoustic models and pronunciation models to multiple accents were based on GMM-HMM based models [3, 4, 7, 5]. Nowadays, deep neural network (DNN) based models are the de-facto standard for acoustic modeling in ASR [8]. To handle accented speech, DNN-based model adaptation approaches have included the use of accent-specific output layers and shared hidden layers [6, 9] and the use of model interpolation to learn accent-dependent models where the interpolation coefficients are learned from data [10]. More recently, an end-to-end based model using the Connectionist Temporal Classification (CTC) loss function was proposed for multi-accented speech recognition [11]. Here, the authors showed that hierarchical grapheme-based models that jointly predicted both graphemes and phonemes performed the best.

Our work is most closely related to very recent work [12] where they jointly learn an accent classifier and a multi-accent acoustic model on American accented speech and British accented speech. Our proposed multi-task architecture is different from their setup which uses separate softmax layers for each accent. We show superior performance on unseen test accents for which no data is available during training.

3. Our Approach

Our proposed approach consists of a multi-task framework where we explicitly supervise a multi-accent acoustic model with accent information by jointly training an accent classifier. Additionally, we train a separate network that learns accent embeddings that can be incorporated as auxiliary inputs within our multi-task framework.

Figure 1 demarcates the three main blocks (A), (B) and (C) that make up our framework:

10.21437/Interspeech.2018-1864

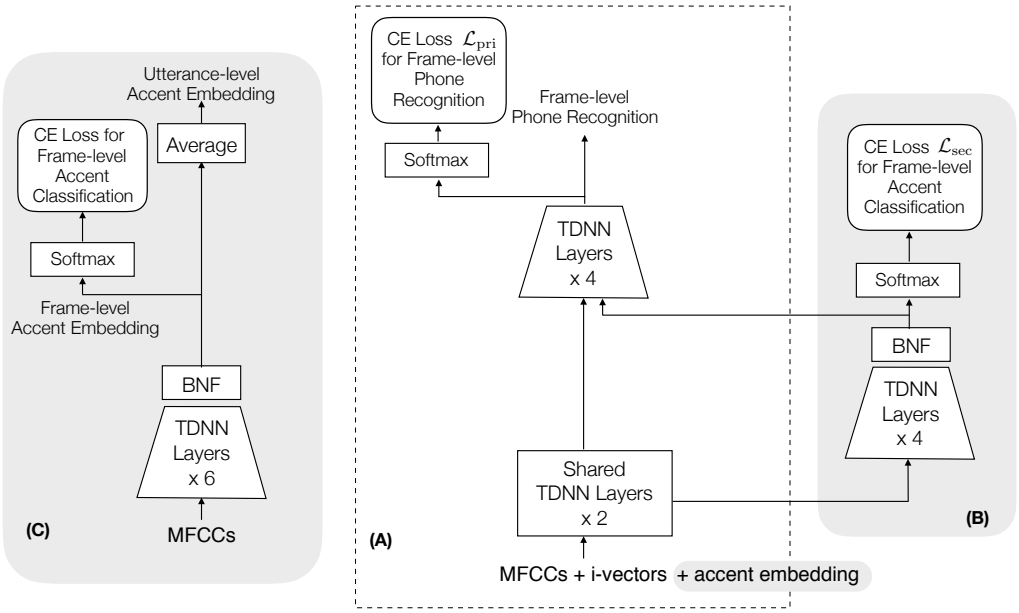

Figure 1: *Multi-task Network Architecture. "CE" refers to cross-entropy loss and "BNF" refers to bottleneck features.*

- Block (A) corresponds to a baseline system that takes as input standard acoustic features (e.g. mel-frequency cepstral coefficients, MFCCs) and i-vector based features that capture speaker information.
- Block (B) is a network trained to classify accents that branches out after two initial shared layers in Block (A) and is trained jointly with the network in Block (A).
- Block (C) is a standalone network that is trained to classify accents. Embeddings from this network can be used to augment the features shown in Block (A).

Choice of neural network units: We chose time-delay neural networks (TDNNs) [13] to build our acoustic model. TDNNs have been demonstrated in prior work to be a more efficient replacement for recurrent neural network based acoustic models [13, 14]. TDNNs are successful in learning long-term dependencies in the input signal using only short-term acoustic features like MFCCs. For these reasons, we adopted TDNNs within our acoustic model. The standalone network in Block (C) used for accent identification is also a TDNN-based model. This was motivated by TDNNs having been successfully used in the past to model long-term patterns in problems related to accent identification such as language identification [15].

Multi-task network: Our multi-task network, illustrated in blocks (A) and (B), jointly predicts context-dependent phone states (which we will refer to as the primary task) and accent labels (which we will refer to as the secondary task). The primary network uses one softmax output layer for context-dependent states across all the accents. (Separate softmax layers for each accent turned out to be a less successful alternative possibly due to varying amounts of speech available across the training accents.) The secondary task has a separate softmax output layer with as many nodes as there are accents in the training data. As input, the secondary task makes use of intermediate feature representations learned from TDNN layers shared across both tasks. Both tasks are trained using separate cross-entropy losses, $\mathcal{L}_{\text{pri}}$ and $\mathcal{L}_{\text{sec}}$, for the primary and secondary networks, respectively.

The secondary softmax layer is preceded by a bottleneck layer of lower dimensionality. These bottleneck features are fed as input to the primary network which predicts context-dependent phones at the frame level. This entire network is trained using backpropagation with the following mixed loss function, $\mathcal{L}_{\text{mixed}}$:

$$\mathcal{L}_{\text{mixed}} = (1-\lambda)\mathcal{L}_{\text{pri}} + \lambda\mathcal{L}_{\text{sec}} \quad (1)$$

where λ is a weight hyperparameter that is used to linearly interpolate the individual loss terms.

During test time, bottleneck features from the secondary network that carry information about the underlying accent are not decoded, but continue to feed into the primary network.

Standalone accent classification network: We also train a standalone TDNN-based accent classifier illustrated in block (C) in Figure 1. The network consists of a bottleneck layer whose activations we refer to as frame-level accent embedding features. These frame-level accent embeddings can be used as auxiliary inputs to network block (A), as shown. Alternately, the vector obtained by averaging across all the frame-level embeddings can serve as an utterance-level accent embedding.

4. Data Description

We use the Common Voice corpus from Mozilla [16] for all our experiments. Common Voice is a corpus of read speech in English that is crowd-sourced from a large number of speakers residing in different parts of the world. (The text comes from various public domain sources like blog posts, books, etc.) Many of the speech clips are associated with metadata including the accent of the speaker (which is self-reported). Across the speech clips annotated with accent information, there are a total of sixteen different accents. We chose seven well-represented accents: United States English (US), England English (EN), Australian English (AU), Canadian English (CA), Scottish English

Dataset	Accents	Hrs of speech	No. of sentences	No. of words
Train-7	US (32), EN (32), AU (14), CA (13), SC (5), IR (3), WE (1)	34.3	30896	283862
Dev-4	US (55), EN (30), AU (8), CA (7)	1.26	1142	10386
Test-4	US (56), EN (27), AU (9), CA (8)	1.25	1127	10467
Test-NZ	NZ	0.59	536	5089
Test-IN	IN	1.33	1200	10780

Table 1: *Statistics of all the datasets of accented speech. Numbers in parenthesis denote the percentage of each accent in the dataset. Number of speakers is same as number of sentences. Train-7 corresponds to training data that is used across all experiments. Dev-4, Test-4, Test-NZ and Test-IN are evaluation datasets; the last two correspond to speech accents that are unseen during training.*

(SC), Irish English (IR) and Welsh English (WE). Our training data ("Train-7") is a mixture of utterances in these seven accents. We constructed a development and test set (referred to as "Dev-4" and "Test-4") using utterances from four of these accents that were disjoint from the training set. As our unseen test accents, we chose New Zealand English and South Asian English (from speakers in India, Pakistan, Sri Lanka) which are denoted as "Test-NZ" and "Test-IN", respectively. We chose these two specific accents as our unseen accents so that we were covering both: 1) a test accent close to one of the training accents (in terms of geographical proximity) i.e. New Zealand English and Australian English and, 2) a test accent sufficiently different from all the training accents i.e. South Asian English. Table 1 shows detailed statistics of these datasets.[1]

5. Experimental Analysis

5.1. Baseline System

All the ASR systems in this paper were implemented using the Kaldi toolkit [17]. Our baseline system was implemented using a feed-forward TDNN network with sub-sampling at intermediate layers. The first layer learns an affine transform of the frames that are spliced together from a window of size $t-2$ to [illegible] s of six T[illegible] ith offset [illegible] ely. Finally, the network consists of an output layer with cross entropy loss across context-dependent phone states. Each TDNN layer has 1024 nodes. Mel-frequency cepstral coefficients (MFCCs), without cepstral truncation, were used as input to the neural network i.e., 40 MFCCs were computed at each time step. Each frame was appended with a 100-dimensional i-vector to support speaker adaptation. We used data augmentation techniques to learn a network that is stable to different perturbations of the data [18]. Three copies of the training data corresponding to speed perturbations of 0.9, 1.0 and 1.1 were created. The alignments used to train this TDNN-based baseline system came from speaker-adapted GMM-HMM tied-state triphone models trained on the "Train-7" data split [19]. A trigram language model was estimated using the training transcripts.

What are the context-dependent phone states?

All network parameters were tuned on "Dev-4" and the best-performing hyperparameters were used for the evaluations on "Test-4", "Test-NZ" and "Test-IN".

[1]Precise details about our data splits are available at: `https://sites.google.com/view/accentsunearthed-dhvani/home`.

Table 2: *Word error rates (WERs) from the multi-task network. Numbers in parentheses denote the interpolation weight of* $\mathcal{L}_{\text{pri}}$.

Model	WER (in %)			
	Dev-4	Test-4	Test-NZ	Test-IN
Baseline	23.1	23.3	24.9	55.2
Multi-task (0.5)	21.3	21.1	27.0	56.9
Multi-task (0.9)	21.2	20.6	23.2	52.1

5.2. Improvements using the Multi-task Network

Table 2 shows the recognition performance using the multi-task network that we described in Section 3 compared to our baseline system. On test data from the first unseen accent, "Test-NZ", the baseline performs reasonably (producing a WER of 25%). However, on test data from the second unseen accent, "Test-IN", the baseline performance is highly degraded and it produces a WER of 55.2%. The interpolation weight for the multi-task network was tuned on "Dev-4": The bests weight were found to be 0.9 for the primary network and 0.1 for the secondary network which we use in multi-task experiments henceforth. We observe from Table 2 that the multi-task network significantly outperforms the baseline system both on seen accents ("Dev-4", "Test-4") and unseen accents ("Test-NZ", "Test-IN").

5.3. Improvements using Accent Embeddings

In Figure 1, we discuss two types of accent embeddings – frame-level and utterance-level – that can be learned from a standalone network and further used as auxiliary inputs during acoustic model training. For the TDNN-based standalone network, we observe that using a 7-layer network with 1024 nodes is preferable to networks with a bottleneck layer of lower dimensionality. Table 3 lists the accent classification accuracy on a validation set (which is created by holding out $1/30^{\text{th}}$ of the training data) by varying the dimensionality of the bottleneck layer from 100 to 1024.

Figures 2 and 3 show the first two PCA dimensions after reducing the dimensionality of utterance-level accent embeddings learned by the standalone network. We include a point for all the utterances in "Dev-4" and color them according to their respective accents. These points are rendered in a lighter shade. It is clear from the figures that these seen accents are fairly well separated. To show where the unseen accented utterances lie, we plot the embeddings of all the utterances in "Test-NZ" in red in Figure 2. Similarly, the embeddings of all the utterances in "Test-IN" are plotted in black in Figure 3. We observe that the unseen accents are grouped together towards the center and appear to share some properties of all the seen accents.

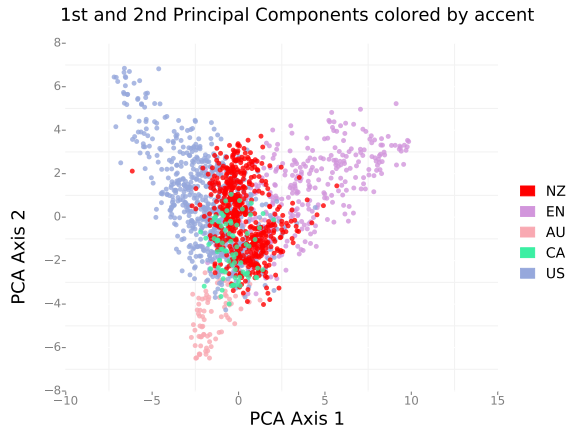

Figure 2: *PCA visualization of utterance-level accent embeddings (from the standalone network shown in block (C) in Figure 1) of the unseen NZ accent + all the accents in Dev-4.*

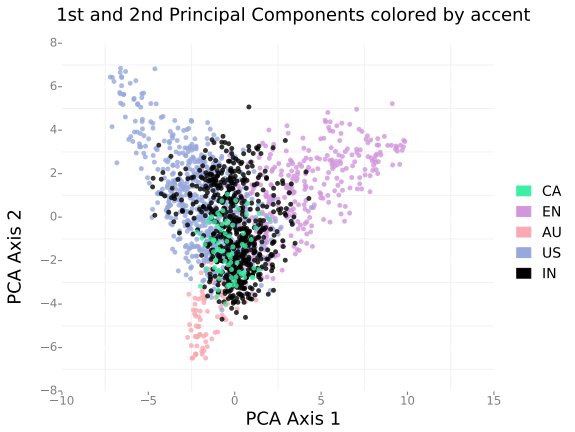

Figure 3: *PCA visualization of utterance-level accent embeddings (from the standalone network shown in block (C) in Figure 1) of the unseen IN accent + all the accents in Dev-4.*

Table 3: *Accent classification accuracy from standalone networks with different bottleneck (BN) layer dimensionalities.*

Model specifications	**Validation acc. (in %)**
TDNN, 7 layers, 100d BN-layer	78.4
TDNN, 7 layers, 200d BN-layer	78.4
TDNN, 7 layers, 300d BN-layer	80.0
TDNN, 7 layers, 1024d BN-layer	82.6

We use the best standalone network from Table 3 with 82.6% validation accuracy to produce frame-level embeddings. These embeddings are averaged across an utterance to obtain a single utterance-level embedding. These embedding features are appended to the MFCC + i-vector features at the frame-level and subsequently used to train the baseline network. Table 4 shows significant improvements over the baseline, across all evaluation sets consisting of seen and unseen accents, by augmenting the input with the accent embeddings during training. This points to the utility of accent embeddings during training; they effectively capture accent-level information (as evidenced in Figures 2 and 3) and make the acoustic model accent-aware. Interestingly, this also has significant impact on the recognition of speech in unseen accents during test time.

5.4. Using Accent embeddings and the Multi-task Network

We finally explore whether there are any benefits from combining both the multi-task architecture along with the accent embeddings learned by the standalone network. The accent embeddings, at the frame-level and the utterance-level, are now fed as auxiliary inputs while training the multi-task network. Table 5 shows recognition performance on all four evaluation sets when the multi-task network is trained using accent embeddings as additional input. Augmenting the input features with accent embeddings improves performance across all four evaluation sets, when compared against the baseline system and the multi-task network in rows 1 and 2, respectively. On both the seen accents ("Dev-4", "Test-4"), we observe statistically significant improvements in WERs (at $p < 0.05$), over the system shown in row 3 in Table 4, when we feed accent embeddings as inputs to the multi-task network. Improvements over the baseline system for all four evaluation sets are statistically significant at $p < 0.001$ using the MAPSSWE test.

Table 4: *Word error rates (WERs) from the baseline network with accent embeddings (AEs) as additional input.*

Model	**WER (in %)**			
	Dev-4	**Test-4**	**Test-NZ**	**Test-IN**
Baseline	23.2	23.3	25	55.2
+ frame-level AEs	21.6	21.8	22.8	50.1
+ utt-level AEs	20.9	21.0	23.0	49.5

Table 5: *Word error rates (WERs) from the multi-task network with accent embeddings (AEs) as additional input. * denotes statistically significant improvements over the baseline at $p < 0.001$ using the MAPSWWE test.*

Model	**WER (in %)**			
	Dev-4	**Test-4**	**Test-NZ**	**Test-IN**
Baseline	23.2	23.3	25	55.2
Multi-task	21.2	20.6	23.2	52.1
+ frame-level AEs	20.4*	20.0*	22.7*	50.9*
+ utt-level AEs	20.0*	19.74*	22.7*	51.2*

6. Conclusions

In this work, we explore the use of a multi-task architecture for accented speech recognition where a multi-accent acoustic model is jointly learned with an accent classifier. Such a network gives far superior performance compared to a multi-accent baseline system, obtaining up to 15% relative WER reduction on a test set with seen accents and 10% relative WER reduction on an unseen accent. Accent embeddings learned from a standalone network give further performance improvements. For future work, we will investigate the influence of accent embeddings when used in multi-accent, end-to-end ASR systems that use recurrent neural network-based models.

7. References

[1] S. Z. L. E. C. C. Huang, T. Chen and J.-L. Zhou, "Analysis of speaker variability," in *Proceedings of Eurospeech*, 2001.

[2] J. J. Humphries, P. C. Woodland, and D. Pearce, "Using accent-specific pronunciation modelling for robust speech recognition," in *Proceedings of ICSLP*, vol. 4. IEEE, 1996, pp. 2324–2327.

[3] Y. Zheng, R. Sproat, L. Gu, I. Shafran, H. Zhou, Y. Su, D. Jurafsky, R. Starr, and S.-Y. Yoon, "Accent detection and speech recognition for Shanghai-accented Mandarin," in *Proceedings of European Conference on Speech Communication and Technology*, 2005.

[4] Y. Liu and P. Fung, "Multi-accent chinese speech recognition," in *Proceedings of ICSLP*, 2006.

[5] H. Kamper and T. Niesler, "Multi-accent speech recognition of afrikaans, black and white varieties of south african english," in *Proceedings of Interspeech*, 2011.

[6] M. Chen, Z. Yang, J. Liang, Y. Li, and W. Liu, "Improving deep neural networks based multi-accent mandarin speech recognition using i-vectors and accent-specific top layer," in *Proceedings of Interspeech*, 2015.

[7] D. Vergyri, L. Lamel, and J.-L. Gauvain, "Automatic speech recognition of multiple accented english data," in *Proceedings of Interspeech*, 2010.

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[9] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-accent deep neural network acoustic model with accent-specific top layer using the kld-regularized model adaptation," in *Proceedings of Interspeech*, 2014.

[10] T. Fraga-Silva, J.-L. Gauvain, and L. Lamel, "Speech recognition of multiple accented english data using acoustic model interpolation," in *Proceedings of EUSIPCO*. IEEE, 2014, pp. 1781–1785.

[11] K. Rao and H. Sak, "Multi-accent speech recognition with hierarchical grapheme based models," in *Proceedings of ICASSP*. IEEE, 2017, pp. 4815–4819.

[12] X. Yang, K. Audhkhasi, A. Rosenberg, S. Thomas, B. Ramabhadran, and M. Hasegawa-Johnson, "Joint modeling of accents and acoustics for multi-accent speech recognition," *arXiv preprint arXiv:1802.02656*, 2018.

[13] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of Interspeech*, 2015.

[14] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Proceedings of Interspeech*, 2015.

[15] D. Garcia-Romero and A. McCree, "Stacked long-term tdnn for spoken language recognition." in *Proceedings of Interspeech*, 2016.

[16] Mozilla, "Project Common Voice," 2017. [Online]. Available: https://voice.mozilla.org/en/data

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[18] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proceedings of Interspeech*, 2015.

[19] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proceedings of ICASSP*, vol. 1, 1992.