Interspeech 2018
2-6 September 2018, Hyderabad

Articulatory and Stacked Bottleneck Features for Low Resource Speech Recognition

Vishwas M. Shetty, Rini A. Sharon, Basil Abraham, Tejaswi Seeram, Anusha Prakash, Nithya Ravi, S. Umesh

Indian Institute of Technology-Madras, India

{ee17s045,ee15d210,ee17d039}@smail.iitm.ac.in, {ee11d032,ee15s044,umeshs}@ee.iitm.ac.in, nithravi1212@gmail.com

Abstract

In this paper, we discuss the benefits of using articulatory and stacked bottleneck features (SBF) for low resource speech recognition. Articulatory features (AF) which capture the underlying attributes of speech production are found to be robust to channel and speaker variations. However, building an efficient articulatory classifier to extract AF requires an enormous amount of data. In low resource acoustic modeling, we propose to train the bidirectional long short-term memory (BLSTM) articulatory classifier by pooling data from the available low resource Indian languages, namely, Gujarati, Tamil, and Telugu. This is done in the context of Microsoft Indian Language challenge. Similarly, we train a multilingual bottleneck feature extractor and an SBF extractor using the pooled data. To bias, the SBF network towards the target language, a second network in the stacked architecture was trained using the target language alone. The performance of ASR system trained with stand-alone AF is observed to be at par with the multilingual bottleneck features. When the AF and the biased SBF are appended, they are found to outperform the conventional filterbank features in the multilingual deep neural network (DNN) framework and the high-resolution Mel frequency cepstral coefficient (MFCC) features in the time-delayed neural network(TDNN) framework.

Index Terms: Articulatory Features, stacked bottleneck, low resource acoustic modeling, BLSTM.

1. Introduction

The work presented in this paper has been carried out as a part of "Low Resource Speech Recognition Challenge for Indian Languages" by Microsoft in Interspeech 2018[1]. This challenge was designed with the objective of developing robust methods to improve the recognition performance of low resource Indian languages.

There are more than 1500 languages in India which are spoken in various dialects across geographical locations. Owing to this diversity, obtaining a sizable amount of data to build good recognition systems for Indian languages is challenging. The state-of-the-art acoustic models require several hours of data for training, especially using discriminative techniques like deep neural networks (DNN). Due to the sparse availability of data, training robust ASR systems for Indian languages requires us to use low resource modeling methods to achieve optimal performance.

[1] www.microsoft.com/en-us/research/event/interspeech-2018-special-session-low-resource-speech-recognition-challenge-indian-languages/

Existing methods for low resource ASR modeling include borrowing data from other high resource languages, pooling/sharing data between similar languages and other model-based approaches. One such model-based approach is the distillation framework where a well-trained teacher network, provides supervision to a student network trained to perform ASR for a low resource language [1]. Another approach to solving the low resource problem is designing better features extractors for the low resource language.

In this paper, we propose a feature extractor module that can generate better feature representations for modeling the low resource speech recognition networks. Here, we explore in detail two types of features, namely, articulatory features (AF) and stacked bottleneck features (SBF). Articulatory features are feature representation of speech signals in terms of the underlying attributes of speech production [2–6]. Hence, they are a form of language-independent modeling as they depend more on the sounds produced by the speaker. The stacked bottleneck (SBN) framework uses a cascade of two networks with bottleneck layers, which provides two levels of abstraction for feature extraction [7]. When these extracted features are pitted against systems built on conventional features such as filter-bank(fbank) or Mel frequency cepstral coefficient (MFCC), these features are seen to improve the recognition performance of low resource languages. Hence, in our method, we propose to append the SBF and AF to form a concatenated feature vector which would be used as the input feature vector for the acoustic model.

The rest of the paper is organized as follows. Details of the data released for the challenge are detailed in Section 2. Articulatory and stacked bottleneck features are reviewed in Sections 3 and 4, respectively. The proposed approach is presented in Section 5. Experiments are detailed in Section 6. Results and performance of the proposed systems are discussed in Section 7. The work is concluded in Section 8.

2. Microsoft Challenge Data

For the challenge, data has been released by SpeechOcean.com and Microsoft. Datasets for Gujarati, Tamil and Telugu have been provided. The data consists of wave files sampled at 16kHz along with corresponding text transcriptions in UTF-8 and a lexicon in terms of the common label set (CLS). Details of the datasets are given in Table1.

3. Articulatory features (AF)

Articulatory features are used in ASR as they incorporate speech production knowledge into ASR [8] and are inherently robust to the speaker and channel variations [9]. Another advantage is the cross-lingual portability of articulatory features

10.21437/Interspeech.2018-2226

Table 1: *Statistics of the data released by Microsoft*

		Train	Dev	Eval
Gujarati	Duration(hrs)	40	5	5
	No. of Utterances	22807	3075	3419
	Average duration(sec)	6.3	5.85	5.26
Telugu	Duration(hrs)	40	5	5
	No. of Utterances	44882	3040	2549
	Average duration(sec)	3.2	5.92	5.92
Tamil	Duration(hrs)	40	5	5
	No. of Utterances	39131	3081	2609
	Average duration(sec)	3.68	5.84	5.88

[10–12]. Hence, articulatory features are better representatives of the articulators that produce speech as compared to phones in each language. Articulatory features model the contextual information in speech better than the conventional features and perform extremely well in adverse acoustic environments [9].

Articulatory features were extracted using the articulatory classifiers trained from acoustic features in [9, 13, 14]. A block schematic of the articulatory feature extraction is shown in Figure 1. Articulatory classifiers need to be constructed for each of the eight articulatory label (AL) groups listed in Table 2. In [15], AF classifiers were trained using the conventional features such as MFCC or filterbank features so as to classify speech signal into these AL groups. Consider a specific example of building an articulatory classifier for the AL group "Degree & Manner". Given an acoustic feature as input, a multilayer perceptron (MLP) is trained with six articulatory labels in "Degree & Manner" group as output targets. This requires the input acoustic features to be aligned at frame-level with the six articulatory labels. [13] showed that manual transcription of data at frame-level in terms of articulatory labels is laborious. Hence, the usual practice is to obtain a phone-level alignment (from an efficient acoustic model built in terms of phones) and convert it into AL using a phone-to-articulatory label (phone-to-AL) mapping. Articulatory features are extracted for each AL group separately and then combined to get the final features as shown in Figure 1. These AFs are called pseudo-AFs. The efficacy of these articulatory features is highly dependent on the amount of speech data available to train articulatory classifiers [13].

The phone-to-AL mapping is not readily available in many of the low-resource languages. In [16], a method to generate phone-to-AL mapping for under-resourced languages was proposed. The idea was to use knowledge from the mapping of a high resourced language. It was based on the center phone capturing property of interpolation vectors obtained from the phone cluster adaptive training (Phone-CAT) method.

In this paper, we use filter bank features to train articulatory classifiers. We also report results using DNN and BLSTM method to train the AF classifiers.

4. Stacked bottleneck (SBN) features

The modeling power of DNNs mainly comes from the complex representations in the hidden layers owing to the availability of a huge amount of training data. A DNN is said to contain a bottleneck (BN) layer if there exists a low dimensional layer which is capable of capturing the abstraction in the data without losing important information required for modeling the data. BN features thus obtained from a network trained on a particular dataset encode higher dimensional multilingual features into a lower dimensional feature set. The encoded features render

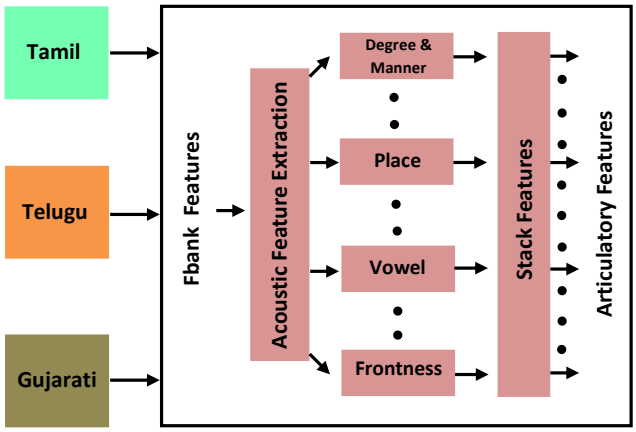

Figure 1: *Articulatory Classifier*

Table 2: *Articulatory feature set*

Group	Cardinality	Feature labels
Glottal	4	aspirated, voiceless, voiced
Place	10	alveolar, dental, labial, labio-dental, lateral, none, post-alveolar, rhotic, velar
Frontness	7	back, front, MID, mid-back, mid-front, nil
Vowel	23	aa, ae, ah, ao, aw1, aw2, ax, ay1, ay2, eh, er, ey, ey1, ey2, ih, iy, ow1,ow2, oy1, oy2, uh, uw, nil
Degree & Manner	6	approximate, closure, FLAP, fricative, vowel
Height	8	HGH, LOW, MID,mid-high, mid-low,very-high, nil
Nasality	3	-, +
Rounding	3	-, +

better feature representations as compared to conventional features.

SBN is made up of two networks, each with a bottleneck layer connected in cascade as shown in Figure 2. The use of hierarchical architectures has shown better performances over regular DNNs as many levels of abstraction are involved to generate better feature representations. The hidden layers in a DNN can be viewed as feature extractors in cascade followed by a logistic regression classifier [17]. Previous works have shown promising results on stacked BN features especially in case of low resource datasets.

In [18], the target language is treated as unseen data and the SBN is trained using the rest of the multilingual data. The network is then biased towards the target language by retraining it using the unseen target language data. In a stacked architecture, the first network is used as the bottleneck feature extractor and the stacked network is used as the target acoustic model [19]. It is observed that the first network in the stacked architecture captures language independent characteristics while the second network is oriented more towards the target language and hence is language dependent [20].

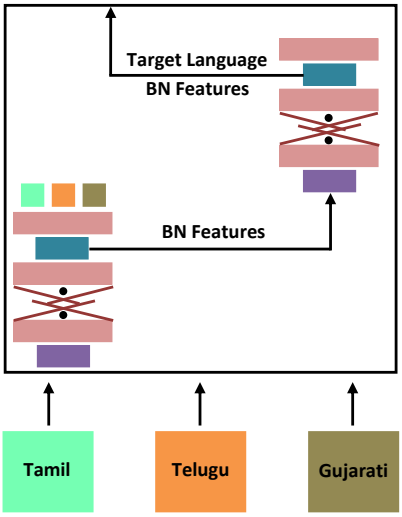

Figure 2: *Stacked Bottleneck Architecture*

Table 3: *Performance of DNN pooled vs BLSTM Pooled AF classifier features (in %WER)*

Language	DNN	BLSTM
Gujarati	17.32	15.21
Telugu	23.80	21.24
Tamil	19.71	19.57

5. Proposed Method

In this paper, we propose a method to generate better feature representation for data in a low resource scenario. We show that significant gains can be obtained by combining AF with the SBN features. AF classifiers model the articulators that produce sound and these features perform extremely well in adverse acoustic conditions. The stack architecture has two levels of DNNs, where the first DNN captures language independent features whereas the second DNN is adapted towards the target language.

Training a robust AF classifier in a low resource scenario can be challenging. In order to account for data scarcity, we pooled the data from all the three available low resource languages and trained a multilingual bidirectional long short-term memory (BLSTM) AF classifier. A common label set was used across all the languages. Experiments were also performed by training articulatory classifiers only on the target language data. It was observed that features from pooled classifiers gave better results than the features from language-specific classifiers. This is due to the fact that the three languages involved being very similar, pooling them makes the parameter estimation for the BLSTM AF classifier network more robust and reliable. Experiments with DNN as AF classifier was also performed. These features didn't perform as well as the BLSTM classifier features. The performance of the two AF classifier features is given in Table 3. Clearly, pooled BLSTM classifier features give the best performance. The results reported here are on the *Dev* set in DNN framework.

The Stacked bottleneck feature extractor used is inspired by the IBM Hierarchical Multilingual DNN that was proposed in [7]. We follow the blocksoftmax approach to train the first network [21]. The first network is a multilingual DNN, trained by pooling data from all the three languages. Each language has its own softmax layer at the output. This network acts as a multilingual feature extractor. Another DNN is built over the first multilingual feature extractor. In order to bias this second network to our target language, we extract bottleneck features for the target language from the bottleneck layer in the first network and train the second network using these features. Thus, we have the initial network capturing language independent features and the second network adapted towards the target language.

When models are trained using a combination of these low resource AF and stacked bottleneck features, a significant improvement was obtained both in the DNN and TDNN framework.

6. Experimental Setup

The experiments were conducted using the Kaldi toolkit [22]. The CDHMM models were trained using 13-dimensional MFCC features with delta and acceleration coefficients. 40-dimensional filter-bank features were used for training the articulatory classifiers and SBN feature extractor.

6.1. AF Extractor

In this section, we describe the AF extraction setup followed in our work. BLSTM classifiers were used as the articulatory classifiers. The articulatory classifiers for each of the AL group were trained as described in Section 3. A common phone-to-AL map manually generated for the pooled data from all the three languages. The AF-BLSTM articulatory classifiers were trained for each of the AL group with 2 layers and cell dimension of 256 using cross-entropy criterion. An initial learning rate of 0.00001 and a momentum of 0.9 was used for training. Once the articulatory classifiers were trained, the articulatory features for the target language were extracted by forward passing its acoustic features through these articulatory classifiers as shown in Figure 1. Then the final articulatory features were obtained by stacking the features from each AL group. The articulatory features were extracted separately for each language to train the language-specific acoustic model.

6.2. SBN feature Extractor

Both the networks in the stack architecture have the same structure. There are 5 hidden layers, which includes one bottleneck layer right before the penultimate layer. All the hidden layers have 1024 nodes each, with the blocksoftmax layer alone having 80 nodes. Experiments were performed with different blocksoftmax dimensions and 80 was found to give the best result. The output layer for the first network comprises of three softmax layers, one for each of the three languages. For a frame of speech from a particular language, cross-entropy is optimized within the posteriors for that language only. At any instant, for a given data point, the output neurons corresponding to the "active language" are only trained whereas all the hidden neurons are trained using backpropagation [21]. From the multilingual trained first network, the bottleneck features for the target language alone are extracted. These features are used to train the second network in the architecture. Unlike the first network, the second network has only one output softmax layer, which is for the target language.

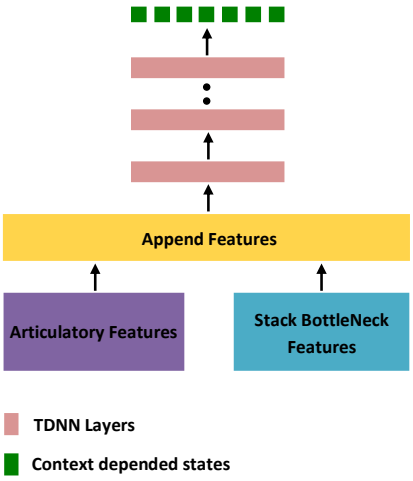

Figure 3: *Final architecture for the method proposed*

6.3. Acoustic Modeling Experiments

In our proposed method, only the feature extraction module is based on a multilingual framework. We train a TDNN based acoustic model for each of the language using hires-MFCC [23], SBN, and articulatory features. The recognition performance of these acoustic models are given in Tables 5 and 6. For each language, the tri-gram language models are trained from the corresponding training data.

6.4. Experiments with Combined Features

The SBN and articulatory features were combined to train acoustic models in each language as shown in Figure 3. The combined features gave improved recognition performance in all cases. The improvements are obtained due to the complementary information present in both features. The recognition performance of the combined features are given in the last column of Tables 5 and 6.

7. Results and Discussions

The results reported in this paper for each language are on the models trained on 40 hours of train data from that language. The baselines reported by Microsoft for the same are given in Table 4. We have reported our results on both the *Dev* set and *Eval* set in Table 5 and Table 6 respectively for different feature inputs. We have shown that the combination of articulatory and SBN features proposed in this paper gives significant improvements over the conventional features in case of Gujarati and Telugu. Our observations from the results reported are the following:

- Pooling data from all the languages to train the articulatory classifiers gave better recognition performance as compared to articulatory classifiers trained on only the target language data.
- Combining the SBN features and articulatory features gave the best recognition performance, thus proving that SBN Feature + AF is indeed a better form of feature representation.
- Surprisingly, just pooling the data from the three languages and training a TDNN network with the hires-MFCC features did not provide gains as expected. The results obtained on *Dev* set were found to be off by an average of 1.50%, when compared to the TDNN hires-MFCC baseline reported in Table 5.
- In Tamil, the best results were obtained with the standalone hires-MFCC features in the TDNN framework.

Table 4: *Baselines by Microsoft on Dev set (in %WER)*

Language	DNN	TDNN
Gujarati	27.79	19.76
Telugu	34.97	22.61
Tamil	25.47	19.45

Table 5: *Resuts on Dev Set (in %WER)*

	DNN	TDNN			
	fbank	hires-MFCC	SBN	AF	SBN+AF
Gujarati	15.08	14.61	14.34	15.15	14.11
Telugu	23.12	21.44	20.19	20.91	19.80
Tamil	18.33	17.32	18.23	19.28	18.16

Table 6: *Results on Eval Set (in %WER)*

	DNN	TDNN			
	fbank	hires-MFCC	SBN	AF	SBN+AF
Gujarati	25.52	24.60	24.82	25.60	24.29
Telugu	34.07	30.40	30.83	30.81	30.33
Tamil	17.93	17.27	18.08	19.03	17.90

Table 7: *Relative improvements obtained by SBN Features + AF in TDNN framework over conventional TDNN and DNN (in %)*

Language	Dev		Eval	
	TDNN	DNN	TDNN	DNN
Gujarati	3.42	6.43	1.26	4.81
Telugu	7.64	14.35	0.23	10.97

8. Conclusions

Various techniques were proposed to improve the acoustic model for all the languages. We have used SBN features and AF for acoustic modeling. We observed that pooling the data from all languages improved the efficacy of the SBN extractor and articulatory classifiers. Finally combining SBN features and AF gave the best performance. A relative improvement of 3.4% and 7.64% was obtained on the Dev sets for Gujarati and Telugu, respectively over hires-MFCC features in TDNN framework. Table 7 shows the relative improvements obtained in TDNN (SBN+AF) over TDNN (hires-MFCC) and DNN (fbank) frameworks.

9. Acknowledgements

The authors would like to thank Microsoft and SpeechOcean.com for conducting the challenge and providing the data.

10. References

[1] Basil Abraham, Tejaswi Seeram, and S Umesh. Transfer learning and distillation techniques to improve the acoustic modeling of low resource languages. In *Proc. Interspeech*, pages 2158–2162, 2017.

[2] Otto Schmidbauer. Robust statistic modelling of systematic variabilities in continuous speech incorporating acoustic-articulatory relations. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 616–619. IEEE, 1989.

[3] Kjell Elenius and G Takács. Phoneme recognition with an artificial neural network. In *Second European Conference on Speech Communication and Technology*, pages 121–124, 1991.

[4] Ellen Eide, J Robin Rohlicek, Herbert Gish, and Sanjoy Mitter. A linguistic feature representation of the speech waveform. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 483–486. IEEE, 1993.

[5] L Deng and D Sun. Phonetic classification and recognition using hmm representation of overlapping articulatory features for all classes of english sounds. In *International Conference on Acoustics, Speech, and Signal Processing*, pages I–45. IEEE, 1994.

[6] Kevin Erler and George H Freeman. An hmm-based speech recognizer using overlapping articulatory features. *The Journal of the Acoustical Society of America*, 100(4):2500–2513, 1996.

[7] Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, Abhinav Sethy, Kartik Audhkhasi, Xiaodong Cui, Ellen Kislal, Lidia Mangu, Markus Nussbaum-Thom, Michael Picheny, et al. Multilingual representations for low resource speech recognition and keyword search. In *Automatic Speech Recognition and Understanding (ASRU)*, pages 259–266. IEEE, 2015.

[8] Simon King, Joe Frankel, Karen Livescu, Erik McDermott, Korin Richmond, and Mirjam Wester. Speech production knowledge in automatic speech recognition. *The Journal of the Acoustical Society of America*, 121(2):723–742, 2007.

[9] Katrin Kirchhoff, Gernot A Fink, and Gerhard Sagerer. Combining acoustic and articulatory feature information for robust speech recognition. *Speech Communication*, 37(3-4):303–319, 2002.

[10] Partha Lal and Simon King. Cross-lingual automatic speech recognition using tandem features. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(12):2506–2515, 2013.

[11] Sunil Sivadas and H Hermansk. On use of task independent training data in tandem feature extraction. In *International Conference on Acoustics, Speech, and Signal Processing*, pages I–541. IEEE, 2004.

[12] László Tóth, Joe Frankel, Gábor Gosztolya, and Simon King. Cross-lingual portability of mlp-based tandem features–a case study for english and hungarian. In *Proc, Interspeech*, 2008.

[13] Joe Frankel, Mathew Magimai-doss, Simon King, Karen Livescu, and Özgür Çetin. Articulatory feature classifiers trained on 2000 hours of telephone speech. In *Proc. Interspeech*, 2007.

[14] Omer Cetin, Amir Kantor, Simon King, Christopher Bartels, Mathew Magimai-Doss, Jorg Frankel, and Karen Livescu. An articulatory feature-based tandem approach and factored observation modeling. In *International Conference on Acoustics, Speech and Signal Processing*, pages IV–645. IEEE, 2007.

[15] Vikramjit Mitra, Ganesh Sivaraman, Hosung Nam, Carol Espy-Wilson, and Elliot Saltzman. Articulatory features from deep neural networks and their role in speech recognition. In *International Conference on Acoustics, Speech and Signal Processing*, pages 3017–3021. IEEE, 2014.

[16] Basil Abraham and Srinivasan Umesh. An automated technique to generate phone-to-articulatory label mapping. *Speech Communication*, 86:107–120, 2017.

[17] A. Ghoshal, P. Swietojanski, and S. Renals. Multilingual training of deep neural networks. In *International Conference on Acoustics, Speech and Signal Processing*, pages 7319–7323. IEEE, 2013.

[18] F. Grézl, M. Karafiát, and K. Veselý. Adaptation of multilingual stacked bottle-neck neural network structure for new language. In *International Conference on Acoustics, Speech and Signal Processing*, pages 7654–7658. IEEE, 2014.

[19] Tanel Alumäe, Stavros Tsakalidis, and Richard M Schwartz. Improved multilingual training of stacked neural network acoustic models for low resource languages. In *Interspeech*, pages 3883–3887, 2016.

[20] E. Chuangsuwanich, Y. Zhang, and J. Glass. Multilingual data selection for training stacked bottleneck features. In *International Conference on Acoustics, Speech and Signal Processing*, pages 5410–5414, 2016.

[21] Karel Veselỳ, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova. The language-independent bottleneck features. In *Spoken Language Technology Workshop*, pages 336–341. IEEE, 2012.

[22] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *Workshop on automatic speech recognition and understanding*. IEEE, 2011.

[23] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.