ARTICULATORY FEATURES FROM DEEP NEURAL NETWORKS AND THEIR ROLE IN SPEECH RECOGNITION

Vikramjit Mitra[1], Ganesh Sivaraman[2], Hosung Nam[3], Carol Espy-Wilson[2], Elliot Saltzman[3,4]

[1]Speech Technology and Research Laboratory, SRI International, Menlo Park, CA
[2]University of Maryland, College Park, MD
[3]Haskins Laboratories, New Haven, CT
[4]Boston University, Boston, MA

ABSTRACT

This paper presents a deep neural network (DNN) to extract articulatory information from the speech signal and explores different ways to use such information in a continuous speech recognition task. The DNN was trained to estimate articulatory trajectories from input speech, where the training data is a corpus of synthetic English words generated by the Haskins Laboratories' task-dynamic model of speech production. Speech parameterized as cepstral features were used to train the DNN, where we explored different cepstral features to observe their role in the accuracy of articulatory trajectory estimation. The best feature was used to train the final DNN system, where the system was used to predict articulatory trajectories for the training and testing set of Aurora-4, the noisy Wall Street Journal (WSJ0) corpus. This study also explored the use of hidden variables in the DNN pipeline as a potential acoustic feature candidate for speech recognition and the results were encouraging. Word recognition results from Aurora-4 indicate that the articulatory features from the DNN provide improvement in speech recognition performance when fused with other standard cepstral features; however when tried by themselves, they failed to match the baseline performance.

Index Terms— *automatic speech recognition, articulatory trajectories, vocal tract variables, deep neural networks.*

1. INTRODUCTION

Spontaneous speech has much variability that poses a significant challenge to the performance of state-of-the-art continuous automatic speech recognition (ASR) systems. A major source of such variability is coarticulation [1] and it has been suggested [2] that coarticulation can be effectively accounted for by incorporating speech production knowledge. A series of studies [3, 4, 5, 7, 11] have demonstrated that articulatory information can improve the performance of ASR systems by systematically accounting for variability such as coarticulation. An indepth exploration of production features and their role in speech recognition systems is provided in [6]. Previous studies [8, 9, 10] have also demonstrated that articulatory representations can help to improve the noise robustness of ASR systems.

In this work we first study a deep neural network (DNN) for estimating articulatory trajectories from speech signals and then use that network to estimate articulatory trajectories for training and testing an English ASR system. DNNs have been successfully used [12, 13] for learning the nonlinear inverse transformation of acoustic waveforms to articulatory trajectories (a.k.a speech inversion). Studies have indicated that articulatory representation can improve phone recognition [13, 15, 16] and speech recognition performance [11, 14]. Studies [13] have demonstrated that use of relevance information (i.e., articulators directly relevant for producing a sound, e.g., tongue tip for /t/) provide lower phone error rates than not using relevance at all.

Speech parameterized in the form of cepstral features was used in this study to train the DNN and we explored four different cepstral features. Based on the results we observed that while the performance of the features differed significantly for smaller nets, but with an increase in the number of hidden layers such differences started to reduce. The DNN was trained using a greedy layer-wise learning procedure where we analyzed the performance of the DNN as more hidden layers were added. The performance of the DNN increased sharply after the first few hidden layers were added and it was found to saturate after the 6th layer, after which no more significant improvement in performance was noted.

Once trained the DNN was used to predict the articulatory trajectories of the training and testing data set of noisy Wall Street Journal (WSJ0) corpus, Aurora-4, and the estimated trajectories were used to train and test an ASR system for an Aurora4 mismatched continuous speech recognition task. We also explored the use of the hidden variables from the DNN as possible candidate features for acoustic model training and the results indicate that they hold significant promise.

The word recognition results on the Aurora-4 task indicate that the use of articulatory information in addition to standard cepstral features provides sufficient complementary information that helps to reduce the word error rates (WER) in noisy and channel-degraded conditions.

2. DATASET FOR DNN TRAINING

To train a model for estimating vocal tract constriction variable trajectories (a.k.a TVs) from speech, we require a speech database containing groundtruth TVs. Currently there are no speech database that contain recorded groundtruth TVs and their corresponding speech waveforms. As a consequence we have used the Haskins Laboratories' Task Dynamic model (popularly known as TADA [17]) along with HLsyn [18] to generate a synthetic English isolated word speech corpus along with TVs. TADA along with HLsyn is an articulatory model based text-to-speech (TTS) converter that given text as input generates vocal tract constriction variables and corresponding synthetic speech. TVs (refer to [9, 10, 20, 21] for more details) are continuous time functions that specify the shape of the vocal tract in terms of constriction degree and location of the constrictors. TADA defines eight TVs altogether as shown in Table 1. Fig. 1 shows the plot of three TVs, LA, TTCD and TBCD, for the utterance 'perfect memory' spoken in a clearly

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

articulated manner. Fig. 1 demonstrates how the TVs behave in the context of different sounds, for example note the dips (denoting constrictions) in TB (tongue body) for /k/, TT (tongue tip) for /t/ and LA (lip aperture) for /m/.

In this work we have used the CMU dictionary [22], 111,929 words were selected and their Arpabet pronunciations were input to TADA. TADA in turn generated their corresponding TVs (refer to Table 1) and synthetic speech. 80% of the data was used as the training set, 10% was used as the cross validation set, and the remaining 10% was used as the test set. Note that TADA generated speech signals at a sampling rate of 8 kHz and TVs at a sampling rate of 200 Hz.

Table 1. Constriction organs and their vocal tract variables.

Constriction organs	vocal tract variables
Lip	Lip Aperture (LA)
	Lip Protrusion (LP)
Tongue Tip	Tongue tip constriction degree (TTCD)
	Tongue tip constriction location (TTCL)
Tongue Body	Tongue body constriction degree (TBCD)
	Tongue body constriction location (TBCL)
Velum	Velum (VEL)
Glottis	Glottis (GLO)

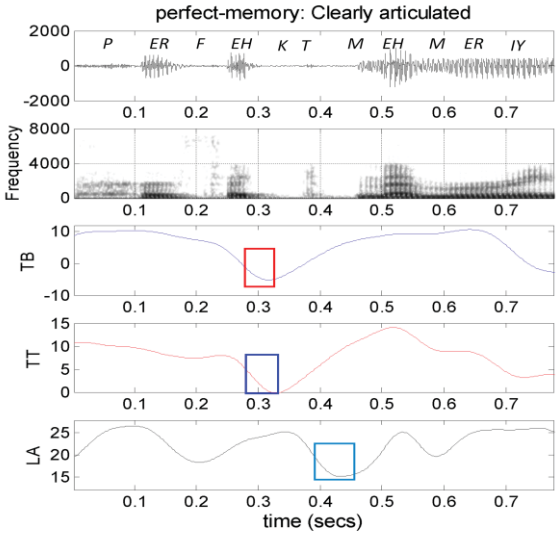

Fig. 1. Waveform, spectrogram and TV plots for TB (tongue body constriction degree: TBCD), TT (tongue tip constriction degree) and LA (lip aperture) for a well-articulated speech saying "perfect memory". The colored squares in the figure demarcates the TB, TT and LA gestures responsible for the /k/, /t/ and /m/ in the utterance.

3. DATASET FOR SPEECH RECOGNITION EXPERIMENTS

For English large vocabulary continuous speech recognition (LVCSR) experiments we used the Aurora-4 noisy Wall Street Journal (WSJ0) dataset. Aurora-4 contains six additive noise versions with channel matched and mismatched conditions. It is created from the standard 5K WSJ0 database and has 7180 training utterances of approximately 15 hours duration and 330 test utterances. The acoustic data (both training and test sets) comes with two different sampling rates (8 kHz and 16 kHz); in our experiments we used only the 8 kHz data. In Aurora-4, two training conditions were specified: (1) clean training, which is the full SI-84 WSJ training set without added noise; and (2) multi-condition training, with about half of the training data recorded using one microphone, and the other half recorded using a different microphone, with different types of added noise at different signal-to-noise ratios (SNRs). The Aurora-4 test data includes 14 test sets from two different channel conditions and six different added noises in addition to the clean condition. The SNR was randomly selected between 0 and 15 dB for different utterances. The six noise types used were: car, babble, restaurant, street, airport and train station. The evaluation set consists of 5K words under two different channel conditions. The original audio data for test conditions 1-7 was recorded with a Sennheiser microphone, while test conditions 8-14 were recorded using a second microphone randomly selected from a set of 18 different microphones (more details in [23]).

4. SPEECH INVERSION - TV ESTIMATION

The task of estimating the TVs from the speech signal is a speech inversion problem, where the acoustic information is used to predict the articulatory trajectories. As an initial experiment with simple artificial neural networks (ANNs) we explored the role of acoustic feature selection in speech inversion.

We explored standard cepstral features such as the traditional Mel-frequency cepstral coefficients (MFCCs) and PLP features using RASTA processing (RASTA-PLP) [24], from which the first 13 cepstral features were used. We also explored acoustic features that are known to be robust to noise and channel degradations: Normalized Modulation Cepstral Coefficient (NMCC) [25] and Synchronized Damped Oscillator Cepstral Coefficients (SyDOCC) [26]. NMCC is a noise robust acoustic feature obtained from tracking the amplitude modulations (AM) of gammatone-filtered subband speech signals in time domain. The AM estimates are obtained using the discrete energy separation algorithm based on the nonlinear Teager's energy operator. The modulation information after root power compression is used to create a cepstral feature, where the first thirteen discrete cosine transform (DCT) coefficients were retained. SyDOCC is a set of perceptually motivated features that represent auditory hair cells as a set of damped oscillators excited by a set of time-synchronized band-limited speech signals. The energy of the oscillator response is transformed using DCT and the first 13 coefficients were retained.

The 13 cepstral coefficients from each of the above four features were Z-normalized. From previous studies [27, 28] we noticed that incorporating contextual dynamic information helps to improve the speech-inversion performance and hence in this work the input features were contextualized by concatenating every other frame within a 200 ms window. This resulted in a feature vector of very large dimension. As a consequence dimensionality reduction was performed on each feature using DCT. We retained the first 70% of the DCT coefficients that resulted in a feature vector of dimension 104. Hence, the neural nets had 104 input neurons and 8 output neurons, corresponding to the eight TVs (see table 1).

4. ASR SYSTEMS

For the Aurora4 ASR experiments, we used SRI International's DECIPHER® LVCSR system, which uses a common acoustic frontend that computes 13 MFCCs (including energy) and their Δs, Δ^2s and Δ^3s. Speaker-level mean and variance normalization is performed on the acoustic features prior to acoustic model training. The acoustic models were trained as crossword triphone HMMs

with decision-tree-based state clustering that resulted in 2048 fully tied states, and each state was modeled by a 32-component Gaussian mixture model. The model uses three left-to-right states per phone and was trained with maximum likelihood estimation. The Aurora-4 system uses the 5K non-verbalized closed vocabulary set language model (LM), where a bigram LM is used in the initial pass of decoding. We performed a second-pass decoding with model space maximum likelihood linear regression (MLLR) speaker adaptation followed by trigram LM rescoring of the lattices. A detailed description of the ASR system is provided in [29].

5. EXPERIMENTS AND RESULTS

5.1 Neural network training

For the initial neural net based TV estimator training we extracted the four acoustic features: MFCC, RASTA-PLP, NMCC and SyDOCC and contextualized them as mentioned in section 4. We initially explored shallower neural nets with up to three hidden layers and compared their performance. The nets were trained with a greedy layer-wise learning, using back propagation with scaled conjugate gradient algorithm. Table 2 below shows the average Pearson's product-moment correlation coefficient (r_{PPMC}) between the actual or ground-truth and the estimated articulatory trajectories (averaged across all the TVs) from neural nets with different numbers of hidden layers. The number of neurons in each of the neural net layers used to obtain table 2 is 150. Note that for a given feature (say NMCC) and a given layer (say layer 1) we trained 3 different neural nets with 150 neurons and if one of those networks was stuck in a local minimum (we analyze this using multiple mini-batches of cross-validation set), then training would restart from the beginning. Table 2 shows that while the selection of acoustic features made some difference in performance for the 1st layer, the differences vanished as the number of layers increased. In our final experiments we used NMCCs as the acoustic feature to train the DNN, the reason behind its selection is that it is a noise robust feature [25] and it demonstrated competitive performance (Table 2) compared to the other features.

Table 2. Average overall r_{PPMC} from neural nets with different number of hidden layers

#Layer	MFCC	RASTA-PLP	NMCC	SyDOCC
1	0.913	0.909	0.901	0.894
2	0.937	0.939	0.937	0.924
3	0.949	0.949	0.950	0.937

For the DNN training we used the greedy layer-wise learning as specified before, where we trained one hidden layer at a time with different numbers of neurons three times and computed the average performance. The number of neurons in each layer was varied from 75 to 200 and the optimal number of neurons was selected based upon the network performance on the cross-validation set (note that the cross validation set and the test set are completely non-overlapping). A final training pass for all the layers was performed after the individual layer-wise learning. We tried pre-training the network but that did not make any difference compared to random initialization of the network. There were altogether six hidden layers with number of neurons - 150, 200, 100, 80, 60 and 40 neurons. We observed that average overall r_{PPMC} increased with addition of hidden layers, i.e., 0.914 after the 1st hidden layer, 0.941 after the 3rd hidden layer and finally became 0.95 after the 6th hidden layer. The networks had tan-sigmoid activation function with a learning rate of 0.01. Table 3 shows the r_{PPMC} values for each of the TVs from the DNN.

Table 3. r_{PPMC} for each TV obtained from the DNN

	GLO	VEL	LA	LP	TTCD	TTCL	TBCD	TBCL
r_{PPMC}	0.956	0.956	0.926	0.938	0.951	0.939	0.946	0.967

5.2 Acoustic model training

The trained DNN was used to generate the TV estimates for Aurora-4 train and test speech data sets. The TV estimates were used in different ways as input to the acoustic model. Our experiments showed the following.

(1) The 8 estimated TVs by themselves are not sufficient for training an acoustic model; the word error rates were very high (~70% at clean) compared to the baseline system.
(2) Combining TVs with MFCC features almost always improved MFCC performance.
(3) Combining TVs with NMCCs showed slight improvement in performance over the NMCC-only system.
(4) The hidden variables from the DNN can also act as features for training an acoustic model, but initial results indicate that their performance is worse compared to the baseline.

We trained an MFCC-based acoustic model as our baseline system. 52 dimensional MFCCs containing 13 cepstral coefficients and their Δs, Δ^2s and Δ^3s were used. The features were computed using an analysis window of 25.6 ms with 10 ms frame rate. We also used the NMCC features [25] in our experiments. The NMCCs used 30 gammatone filterbanks equally spaced in the ERB scale between 250 to 3750 Hz and analyzed speech using a 25.6 ms hamming window with 10 ms frame rate. They consists of 13 cepstral features which were concatenated with their Δs, Δ^2s and Δ^3s, resulting in a 52 dimensional feature set. For the Aurora-4 ASR experiments we used only the mismatched conditions (i.e., train with clean data and test on data from different noisy backgrounds and the same or different channels) at 8 kHz. We used mismatched condition training as it is harder than the multi-condition training in Aurora-4 experiments.

We first tried using the hidden variables from the first three hidden layers of the DNN as observations for training the acoustic model. Note that the dimension of these hidden variables was very high, so we performed principal component analysis (PCA) on them and reduced their dimension to 40, where we observed that more than 90% of the information was retained within these forty dimensions. Table 4 shows the word error rates for the mismatched channel testing condition of Aurora-4.

Table 4 shows an interesting trend, where the WERs increased as a higher layer is chosen to train an acoustic model. Note that DNN performs a series of nonlinear transformations (owing to the nonlinear activation function sandwiched between the layers) of the acoustic feature to generate the TVs. So within the first few layers the result of such nonlinear transformation would generate a space more similar to the acoustic space and then as we propagate through the layers that space becomes more warped to the output TV space. As we know from our previous experiments [11] and also from the current experiment, the articulatory features by themselves are not complete and sufficient to train a full blown acoustic model, hence we can assert that the articulatory space learnt through the DNN is also not complete for acoustic model training. This may justify why the WERs increase as we traverse the DNN pipeline from left to right. However note that the hidden variables from layers 1 and 2 did improve the WER slightly compared to the baseline system for street and train noise conditions.

In our next set of experiments we explored the use of TVs by themselves and in their time contextualized form. The latter provided better performance as the dynamic information of the TV trajectories seems to have a role in the performance of the articulatory features. The contextualization of TVs was performed

Table 4. WER for clean training mismatched channel condition

		MFCC	Hidden-1	Hidden-2	Hidden-3
1	Clean	**15.0**	27.0	27.3	38.2
2	Car	**20.6**	33.2	34.0	46.1
3	Babble	**44.7**	47.7	49.7	58.6
4	Restaurant	**48.3**	53.9	55.0	68.0
5	Street	52.9	**51.3**	51.9	62.8
6	Airport	**39.8**	48.3	51.0	61.2
7	Train	51.4	51.1	**50.3**	61.4
	Avg. (2-7)	42.9	47.6	48.7	59.7

using a context of 13 frames that contained ~120 ms of temporal information. This yielded a feature vector of 104 dimensions. To reduce the dimension of the contextualized TVs, DCT was performed on each of the eight TV dimensions and their first seven coefficients were retained, resulting in a 56 dimensional feature, which captures the modulation of TVs (and called them the ModTVs). We trained acoustic models with only the 8 TVs, the 56 dimensional modTVs and the modTVs after heteroscedastic linear discriminant analysis (HLDA) based transform to 10 dimensions (modTV_hlda10). The modTVs provided better results followed by modTV_hlda10 and TVs, indicating that dynamic information is useful for ASR. In all these experiments the results from using only the articulatory features were much worse than the baseline system and hence those results are not reported here.

From our previous work [9, 11] we learnt that the articulatory features work the best when they are combined with one of the standard acoustic features. We concatenated the 56 dimensional ModTV features with 52 dimensional MFCC features, resulting in a 108 dimensional feature set. We then performed PCA on these 108 dimensional MFCC+ModTV features and noticed that the first 30 dimensions retain more than 90% of the information. Based on this, we reduced the dimension of 108 dimensional MFCC+ModTV features to 30 and call this feature MFCC+ModTV_pca30.

As an alternative we also fused the ModTV features with 52 dimensional NMCC features using the same approach as the MFCCs. PCA was performed on the resulting features and we noticed that the meaningful information resided within the top 30 dimensions after PCA transform. We call the resulting feature NMCC+ModTV_pca30. Tables 5 and 6 show the WERs obtained from the baseline (MFCC) system, NMCC system and the systems trained with MFCC+ModTV_pca30 and NMCC+ModTV_pca30 features.

Tables 5 and 6 show that the articulatory features helped to reduce the relative WER by 6.8% and 2.6% under matched and mismatched channel clean condition for MFCC, and by 3.1% and 1.1% under the same conditions for NMCC. They also helped to reduce the WER in noisy conditionz. The effectiveness of the articulatory features is more pronounced under mismatched channel conditionz where they reduced the relative overall WER by 9.1% and 2.2% for MFCC and NMCC respectively. While under matched channel condition we did not observe any reduction in WER for NMCC+ModTV_pca30 features compared to the NMCCs we did observe a relative overall WER reduction 9.8% in the MFCC+ModTV_pca30 features compared to the MFCCs. Note that we have also explored HLDA instead of PCA for reducing the dimensions of the MFCC+ModTV and NMCC+ModTV features, but the results were not as promising as the PCA ones. We also explored reducing the dimension of the MFCC baseline features to 30 using HLDA (MFCC_hlda30) in order to have a direct comparison with the 30 dimensional MFCC+ModTV_pca30 features. We observed that the MFCC_hlda30 demonstrated an overall relative WER reduction of 4.4% compared to the MFCCs, but the MFCC+ModTV_pca30 still gave lower WERs than the MFCC_hlda30 features. The relative WER reduction from MFCC+ModTV_pca30 features were 6.4% and 6.5% under matched and mismatched noisy conditions respectively, compared to the MFCC_hlda30 features.

Table 5. WER for clean training matched channel condition

		MFCC	MFCC+ModTV pca30	NMCC	NMCC+ModTV_ pca30
1	Clean	11.7	**10.9**	13.4	13.0
2	Car	16.6	**16.5**	17.3	18.3
3	Babble	37.9	33.8	**32.6**	32.9
4	Restaurant	41.5	37.6	**35.3**	37.0
5	Street	45.1	38.8	34.7	**33.5**
6	Airport	33.2	29.9	**30.1**	30.6
7	Train	45.7	42.0	34.8	**33.9**
	Avg. (2-7)	36.7	33.1	30.8	31.0

Table 6. WER for clean training mismatched channel condition

		MFCC	MFCC+ModTV pca30	NMCC	NMCC+ModTV_ pca30
1	Clean	15.0	**14.6**	17.4	17.2
2	Car	20.6	**19.9**	21.7	20.8
3	Babble	44.7	39.7	37.0	**35.1**
4	Restaurant	48.3	43.9	**41.4**	42.1
5	Street	52.9	47.9	40.3	**40.2**
6	Airport	39.8	35.6	35.7	**34.3**
7	Train	51.4	46.7	39.2	**38.3**
	Avg. (2-7)	42.9	39.0	35.9	35.1

6. CONCLUSION

In this work we presented a DNN for estimating articulatory trajectories from the speech signal and demonstrated that with deeper networks we can improve the performance of the TV estimation compared to shallower nets. We explored different acoustic features for speech inversion and noticed that for deeper nets the selection of features does not have a major impact. We also demonstrated that the hidden variables from the deep net can be used as acoustic features; however their performance was not up to the mark compared to the baseline system; this is expected as the network was trained with clean synthetic speech. Also another constraint in our DNN training was the fact that the articulatory data used to train the DNN was obtained from a single speaker model. In future we intend to use data artificially corrupted with different noise types as well as different speaker models to train the DNN. In such a case the hidden variables from the DNN can be a competitive candidate compared to the baseline. We are also exploring ways by which we can transform recorded articulatory data in the form of pellet (or flesh-point) trajectories to TV (constriction variables) trajectories and hence train the DNN using natural speech. Our results indicate that though articulatory features by themselves may not be a standalone feature for speech recognition, when combined with other features they help in improving speech recognition performance under clean as well as noise/channel degraded conditions.

7. ACKNOWLEDGMENTS

This research was supported by NSF Grant # IIS-0964556, IIS-1162046 and IIS-1161962.

8. REFERENCES

[1] R. Daniloff and R. Hammarberg, “On defining coarticulation”, *J. of Phonetics*, Vol.1, pp. 239-248, 1973.

[2] K. N. Stevens, “Toward a model for speech recognition”, *J. of Acoust. Soc. Am.*, Vol.32, pp. 47-55, 1960.

[3] K. Kirchhoff, *Robust Speech Recognition Using Articulatory Information*, PhD Thesis, University of Bielefeld, 1999.

[4] J. Frankel and S. King, “ASR - Articulatory Speech Recognition”, *Proc. of Eurospeech*, pp. 599-602, Denmark, 2001.

[5] L. Deng and D. Sun, “A statistical approach to automatic speech recognition using atomic units constructed from overlapping articulatory features”, *J. of Acoust. Soc. Am.*, 95(5), pp. 2702-2719, 1994.

[6] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond and M. Wester, "Speech production knowledge in automatic speech recognition", *J. Acoust. Soc. of Am.*, 121(2), pp. 723-742, 2007.

[7] K. Livescu, O. Cetin, M. Hasegawa-Johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, L. Yung, A. Bezman, S. Dawson-Haggerty, B. Woods, J. Frankel, M. Magimai-Doss and K. Saenko, “Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU Summer Workshop”, *Proc. of ICASSP*, Vol.4, pp.621-624, 2007.

[8] M. Richardson, J. Bilmes and C. Diorio, “Hidden-articulator Markov models for speech recognition”, *Speech Comm.*, 41(2-3), pp. 511-529, 2003.

[9] V. Mitra, H. Nam, C. Espy-Wilson, E. Saltzman and L. Goldstein, “Articulatory information for noise robust speech recognition”, *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 19, Iss. 7, pp. 1913-1924, 2010.

[10] V. Mitra, H. Nam and C. Espy-Wilson, “Robust speech recognition using articulatory gestures in a Dynamic Bayesian Network framework”, *Proc. of Automatic Speech Recognition & Understanding Workshop*, ASRU, pp. 131-136, Hawaii, 2011.

[11] V. Mitra, W. Wang, A. Stolcke, H. Nam, C. Richey, J. Juan, and M. Liberman, “Articulatory features for large vocabulary speech recognition,” in Proc. IEEE ICASSP, Vancouver, May 2013.

[12] B. Uria, S. Renals, and K. Richmond, “A Deep Neural Network for Acoustic-Articulatory Speech Inversion”, *in NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning,* 2011.

[13] Canevari, C., Badino, L., Fadiga, L., Metta, G., ”Relevance-weighted reconstruction of articulatory features in Deep Neural Network-based Acoustic-to-Articulatory Mapping”, *in Proc. of Interspeech*, 2013.

[14] V. Mitra, H. Nam, C. Espy-Wilson, E. Saltzman, L. Goldstein, “Gesture-based Dynamic Bayesian Network for Noise robust Speech Recognition,” *in Proc. of ICASSP*, pp. 5172-5175, 2011.

[15] L. Deng and D. Sun, “A statistical approach to ASR using atomic units constructed from overlapping articulatory features”, *J. of Acoust. Soc. Am.*, 95, pp. 2702–2719, 1994.

[16] K. Erler and L. Deng, “Hidden Markov model representation of quantized articulatory features for speech recognition”, *Comp., Speech & Lang.*, Vol. 7, pp. 265–282, 1993.

[17] H. Nam, L. Goldstein, E. Saltzman and D. Byrd, “Tada: An enhanced, portable task dynamics model in Matlab”, *J. of Acoust. Soc. Am.*, 115(5), pp. 2430, 2004.

[18] H. M. Hanson and K. N. Stevens, “A quasiarticulatory approach to controlling acoustic source parameters in a Klatt-type formant synthesizer using HLsyn”, *J. of Acoust. Soc. Am.*, 112(3), pp. 1158-1182, 2002.

[19] C. P. Browman and L. Goldstein, “Towards an Articulatory Phonology”, *Phonology Yearbook*, 85, pp. 219-252, 1986.

[20] C. P. Browman and L. Goldstein, “Articulatory Phonology: An Overview”, *Phonetica*, 49, pp. 155-180, 1992.

[21] E. Saltzman and K. Munhall, “A Dynamical Approach to Gestural Patterning in Speech Production”, *Ecological Psychology*, 1(4), pp. 332-382, 1989.

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

[23] G. Hirsch, “Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task”, *ETSI STQ-Aurora DSR Working Group*, June 4, 2001.

[24] H. Hermansky and N. Morgan, “RASTA processing of speech,” IEEE Trans. Speech Audio Proc., vol.2, pp.578–589, 1994.

[25] V. Mitra, H. Franco, M. Graciarena and A. Mandal, “Normalized amplitude modulation features for large vocabulary noise-robust speech recognition”, Proc. IEEE CASSP, pp. 4117-4120, 2012.

[26] V. Mitra, H. Franco, M. Graciarena, “Damped Oscillator Cepstral Coefficients for Robust Speech Recognition,” Proc. Interspeech, pp. 886-890, 2013.

[27] V. Mitra, H. Nam, C. Espy-Wilson, E. Saltzman and L. Goldstein, “Retrieving tract variables from acoustics: a comparison of different machine learning strategies”, *IEEE Journal of Selected Topics on Signal Processing*, Sp. Iss. on Statistical Learning Methods for Speech and Language Processing, Vol. 4, Iss. 6, pp. 1027-1045, 2010.

[28] K. Richmond, “Estimating Articulatory parameters from the Acoustic Speech Signal”, *PhD Thesis*, Univ. of Edinburgh, UK, 2001.

[29] A. Stolcke, B. Chen, H. Franco, V. R. R. Gadde, M. Graciarena, M.-Y. Hwang, K. Kirchhoff, A. Mandal, N. Morgan, X. Lin, T. Ng, M. Ostendorf, K. Sonmez, A. Venkataraman, D. Vergyri, W. Wang, J. Zheng and Q. Zhu, “Recent Innovations in Speech-to-Text Transcription at SRI-ICSI-UW”, *IEEE Trans. on Audio, Speech and Language Processing*, 14(5), pp. 1729-1744, 2006.

[30] A. Mandal, M. Ostendorf and Andreas Stolcke, “Improving robustness of MLLR adaptation with speaker-clustered regression class trees”, *Computer Speech & Language*, 23, pp. 176 199 (2009). ISSN 0885-2308.