INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Accent Identification by Combining Deep Neural Networks and Recurrent Neural Networks Trained on Long and Short Term Features

Yishan Jiao[1], *Ming Tu*[1], *Visar Berisha*[1,2], *Julie Liss*[1]

[1]Department of Speech and Hearing Science
[2]School of Electrical, Computer, and Energy Engineering
Arizona State University
{yjiao16, mingtu, visar, julie.liss}@asu.edu

Abstract

Automatic identification of foreign accents is valuable for many speech systems, such as speech recognition, speaker identification, voice conversion, etc. The INTERSPEECH 2016 Native Language Sub-Challenge is to identify the native languages of non-native English speakers from eleven countries. Since differences in accent are due to both prosodic and articulation characteristics, a combination of long-term and short-term training is proposed in this paper. Each speech sample is processed into multiple speech segments with equal length. For each segment, deep neural networks (DNNs) are used to train on long-term statistical features, while recurrent neural networks (RNNs) are used to train on short-term acoustic features. The result for each speech sample is calculated by linearly fusing the results from the two sets of networks on all segments. The performance of the proposed system greatly surpasses the provided baseline system. Moreover, by fusing the results with the baseline system, the performance can be further improved.

Index Terms: accent identification, deep neural networks, prosody, articulation

1. Introduction

Accent classification refers to the problem of inferring the native language of a speaker from his or her foreign accented speech. Identifying idiosyncratic differences in speech production is important for improving the robustness of existing speech analysis systems. For example, automatic speech recognition (ASR) systems exhibit lower performance when evaluated on foreign accented speech. By developing pre-processing algorithms that identify the accent, these systems can be modified to customize the recognition algorithm to the particular accent [1] [2]. In addition to ASR applications, accent identification is also useful for forensic speaker profiling by identifying the speaker's regional origin and ethnicity in applications involving targeted marketing [3] [4]. In this paper we propose a method for classification of 11 accents directly from the speech acoustics.

A number of studies have analyzed how elemental components of speech change with accent. Spectral features (e.g. formant frequencies) and temporal features (e.g. intonation and durations) have all been shown to vary with accent [5] [6]. These features have been combined in various statistical models and machine learning methods to automate the accent classification task. Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) are commonly used approaches in many earlier studies [7] [8] [9]. For example, Deshpande et al. used GMMs based on formant frequency features to discriminate between standard American English and Indian accented English [7]. Chen et al. explored the effect of the number of components in GMMs on classification performance [10]. Tang and Ghorbani compared the performance of HMMs with Support Vector Machine (SVM) for accent classification [11]. Others have also considered linear models. Ghesquiere et al. used both formant frequencies and duration features and proposed an "eigenvoice" approach for Flemish accent identification [8]. Kumpf and King proposed to use linear discriminant analysis (LDA) for identification of three accents in Australian English [12].

Artificial neural networks, especially Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs) have been widely used in state-of-the-art speech systems [13] [14] [15] [16]; however in the area of accent identification, there are only a few studies evaluating the performance of neural networks [17] [18]. Nonetheless, in a related area, language identification (LID), neural networks have been investigated exhaustively [19] [20] [21]. A recent study in this area explored the use of recurrent neural networks for automatic language identification [22]. Their study also suggests that the combination of recurrent and deep networks can lead to significant improvements in performance. Inspired by this work, in this paper, we propose a system that combines DNNs and RNNs. In contrast to the work in [22], we propose to take advantage of both long-term and short-term features since previous work shows that foreign accents depend on both long-term *prosodic features* and short-term *articulation features*. The final prediction is obtained by linearly fusing the results from the two neural networks.

The organization of this paper is as follows. Section 2 briefly describes the goal, the dataset and the baseline system for the INTERSPEECH 16 Native Language Sub-Challenge. Section 3 introduces the proposed system that combines long and short term features using DNNs and RNNs. The corresponding experimental setup is also described in this section. The evaluation results are shown in Section 4. The discussion and the conclusion are in Section 5.

2. Dataset and the Baseline System

The provided dataset for the INTERSPEECH 16 Native Language Sub-Challenge contains a training, a development, and a test set. The corpus contains one speech sample from 5132 speakers, labeled with one of the 11 native languages. The training and development sets are each assigned 3300 and 965 samples respectively. The remaining 867 samples are assigned to the test set. The length of each sample is 45 seconds. A detailed description of the dataset can be found in the baseline

http://dx.doi.org/10.21437/Interspeech.2016-1148

Table 1: Confusion matrix of baseline system on development set. Rows are reference, and columns are hypothesis.

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR
ARA	**31**	3	6	7	5	5	6	5	5	6	7
CHI	4	**38**	5	4	5	2	5	9	7	4	1
FRE	11	7	**29**	9	0	5	3	1	9	0	6
GER	4	4	5	**54**	1	7	2	3	6	1	0
HIN	3	2	2	0	**48**	2	1	2	2	21	0
ITA	6	3	8	7	6	**46**	0	3	10	1	4
JPN	4	13	5	2	2	1	**35**	11	10	1	1
KOR	3	20	1	3	2	3	13	**31**	5	3	6
SPA	6	11	15	6	2	4	9	8	**33**	1	5
TEL	2	0	3	2	24	2	3	1	2	**42**	2
TUR	6	4	4	6	2	6	7	8	5	0	**47**

paper [23].

The goal of the Native Language Sub-Challenge is to identify the corresponding native language from the accented speech. The challenge is particularly difficult for two reasons: first, all of the speech samples were recorded with babel background noise using low-quality head mounted microphones. Second, in addition to accent differences, a large number of the speakers were not perfectly fluent in English; therefore there were a number of pauses and linguistic fillers in the speech. In our proposed system we try to address these challenges by using a voice activity detection (VAD) to remove the pauses and using a non-linear learning algorithm to model the relationship between the features and the class label.

The baseline system against which we compare used 6373 long-term features extracted from each speech sample with openSMILE [24]; these include prosodic features (range, maximum, minimum of F0, sub-band energies, peaks, etc.) and various statistics of traditional acoustic features (mean, standard deviation, kurtosis of MFCC, RASTA, etc.). A support vector machine (SVM) is constructed to model the data. More detail about the baseline system can be found in [23]. The performance of the baseline system on development set is shown as a confusion matrix in Table 1. The overall accuracy is 44.66%. The recall for each class and the unweighted average recall (UAR) is shown in the second column of Table 2.

3. Proposed System Description and Experimental Setup

The proposed system is shown in Figure 1. It consists of a voice activity detector, followed by two parallel neural networks (a DNN and an RNN) analyzing the speech samples at different scales, and a probabilistic fusion algorithm. Below we describe each component of the model.

Voice Activity Detection: As mentioned previously, there are a number of pauses and silences in the speech samples. These were often due to the fact that some of the speakers did not speak fluent English and paused to think of the proper expression. We first used voice activity detection (VAD) [25] to remove the silence periods. The VAD threshold was adjusted to match the noise level of the speech samples using cross-validation and we only removed the detected silence segments with length longer than 300 milliseconds.

Framing and Feature Extraction: The remaining speech samples were then trimmed into multiple segments with equal length of 4 seconds. Thus every 45-second speech sample was segmented into approximately 10-11 parts. Long-term features we used were the same as those in the baseline system (mean, standard deviation, kurtosis of MFCC, RASTA, etc.). They were extracted from each segment in each speech sample with openSMILE scripts. Each 4-sec window was further split into 25ms windows with a 10ms overlap. Short-term features were extracted from each 25ms signal. Specifically, we used 39^{th}-order mel-scale filterbank features with logarithmic compression [26].

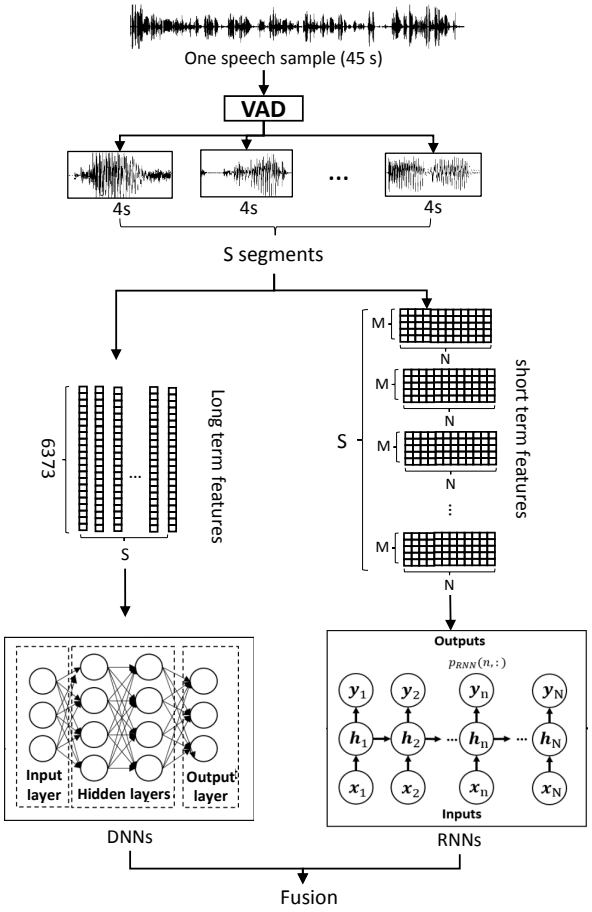

Figure 1: The proposed system of combining long and short term features using DNNs and RNNs.

Deep Neural Network: A DNN was constructed to make a prediction regarding the accent type from the long-term features. The structure of the DNN is as follows: There was an input layer with 6373 nodes corresponding to each dimension in the feature set. Three hidden layers with 256 nodes for each followed. Rectifier linear units ("ReLU") were used at the output of each layer and we use the dropout method to prevent overfitting - each input unit to the next layer can be dropped with 0.5 probability [27]. The output layer contained 11 nodes corresponding to the 11 accents with softmax activation functions. Stochastic gradient descent with a batch size of 128 was used for training. The learning rate and momentum were set to 0.001 and 0.9 respectively. All of the parameters were optimized on the development set. We attempted to use principal component analysis (PCA) to reduce the input feature dimension from

Table 2: Recall for each class and the unweighted average recall (UAR) on development set given by different systems (%)

	Baseline	DNN+RNN	Baseline +DNN+RNN
ARA	36.0	39.5	41.9
CHI	45.2	65.4	65.5
FRE	36.3	45.0	50.0
GER	62.1	62.4	68.2
HIN	57.8	79.5	77.1
ITA	48.9	64.9	68.1
JPN	41.2	43.5	44.7
KOR	34.4	42.2	47.8
SPA	33.0	26.0	35.0
TEL	50.6	43.4	49.4
TUR	49.5	62.8	66.0
UAR	**45.1**	**52.2**	**55.8**

6373 to 800. Our hope was that this would reduce the size of the model, making it easier to train, and improving its robustness; however the cross-validation results on the development set after PCA decreased slightly, therefore we kept the original feature set.

Recurrent Neural Network: The RNN was trained on the short-term features extracted from 25ms frames of speech. Categorical labels were assigned to each frame of the segment. The results for each sample were calculated by averaging the predictions on all frames in all segments. The structure of RNN is as follows: The input data is sequentially fed into the RNN frame-by-frame. Each frame is of dimension 39. Two hidden layers with 512 long short term memory (LSTM) nodes were used. In each LSTM node, there is a cell state regulated by a forget gate, an input gate and an output gate. The activation function for the gates was a 'logistic sigmoid' and for updating the cell state we used a 'tanh'. The accent label was assigned to every 25ms speech frame - the LSTM layers allowed the model to learn long-term dependencies by taking the output of the previous hidden nodes as part of the inputs to the current nodes. Our hypothesis was that with this kind of structure the model could learn differences in articulation (e.g. formant values) and differences in how articulation changes over time (e.g. formant trajectories) for different accents. Specifically, as shown in the RNN part of Figure 1, the input is a time series of acoustic features $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n, ..., \mathbf{x}_\mathrm{N}]$ with length N. After training, the RNN computes the hidden sequences $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_n, ..., \mathbf{h}_\mathrm{N}]$ and outputs the probability predictions for each frame $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n, ..., \mathbf{y}_\mathrm{N}]$ by iterating from $n = 1$ to N as follows [28]:

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_t &= f_{\boldsymbol{\theta}}(\mathbf{W}_{x\overrightarrow{h}}\mathbf{x}_t + \mathbf{W}_{\overrightarrow{h}\overrightarrow{h}}\mathbf{h}_{t-1} + \mathbf{b}_{\overrightarrow{h}}), \\ \mathbf{y}_t &= \mathbf{w}_{\overrightarrow{h}y}\overrightarrow{\mathbf{h}}_t + \mathbf{b}_y. \end{aligned} \tag{1}$$

For training the model, we followed a similar approach to the DNN. We used the dropout methods with each of the input units to the next layer dropped in 0.5 probability [29]. The RMSProp algorithm was used for optimization [30] with a learning rate of 0.001 and a batch size for training of 256 samples.

Generating a final decision: We interpret the output of the activation functions of both the DNN and the RNN as a posteriori probabilities. The final decision was calculated by fusing these two estimations. Suppose the complete speech sample from a speaker ($\approx$ 45 sec) is segmented into S 4-sec parts. The DNN provides as an output a probability vector that describes the probability that the input segment belongs to any of the 11 classes. Thus, the probability prediction given by DNN for the i^{th} segment in the j^{th} class is denoted by $P_{\mathrm{DNN}}(i,j)$, where $i = 1, 2, ..., S$ and $j = 1, 2, ..., 11$. The RNN also provides a probability vector, but it is predicted on every 25ms frame instead of on every segment. For the same 4-sec segment used in the DNN, we can combine the results from the individual frames into a single prediction for the segment, $P_{\mathrm{RNN}}(i,j)$, as follows:

Table 3: Accuracy and UAR for the variations of the systems on development set

	RNN only	DNN only	Fusion on segments	DNN with RNN(on sequence)
Accuracy (%)	42.9	49.1	49.8	50.2
UAR (%)	43.2	49.5	50.0	50.4

$$P_{\mathrm{RNN}}(i,j) = \frac{1}{N}\sum_{n=1}^{N} p_{\mathrm{RNN}}(n,j) \tag{2}$$

where $P_{\mathrm{RNN}}(n,j)$ is the prediction of the RNN on the n^{th} frame for the j^{th} class dimension with $n = 1, 2, ...N$, $j = 1, 2, ..., 11$. N is the total number of frames in speech segment i, $i = 1, 2, ..., S$.

After combining the individual probabilities for each frame into a single probability for the segment, we can combine the DNN and RNN probabilities using a weighted average. The final probability score $P(j)$ on the complete sample in the j^{th} class is calculated as Equation 3.

$$P(j) = \frac{1}{S}\left[w_{\mathrm{DNN}}\sum_{i=1}^{S} P_{\mathrm{DNN}}(i,j) + w_{\mathrm{RNN}}\sum_{i=1}^{S} P_{\mathrm{RNN}}(i,j)\right] \tag{3}$$

where $i = 1, 2, ...S$, $j = 1, 2, ..., 11$. w_{DNN} and w_{RNN} are the weights for DNN and RNN predictions. They are determined by the accuracy of DNN and RNN on the development set as follows,

$$\begin{cases} w_{DNN} = \frac{Acc_{DNN}}{Acc_{DNN}+Acc_{RNN}} \\ w_{RNN} = 1 - w_{DNN} \end{cases} \tag{4}$$

where Acc_* is the accuracy of the model, which is the proportion of correct predictions. A final decision is made by selecting the class with the highest probability.

4. Evaluation and Results

Both the DNN and RNN were trained with the Python neural networks library, Keras [31], running on top of Theano on a CUDA GPU. The data was normalized to zero mean and unit standard deviation, using the mean and standard deviations from the training set. The results are shown as recall for each class in the third column of Table 2. The overall accuracy is 51.92%, and the UAR is 52.24%.

We also made a number of variations of the system and tested the performance on the development set. The first two variations (DNN only and RNN only) use the DNN and RNN alone without any fusion. The third variation (Fusion on segments) uses both the DNN and RNN, but the prediction is obtained by

Table 4: Confusion matrix of the proposed system fused with baseline on development set. Rows are reference, and columns are hypothesis.

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR
ARA	**36**	3	3	8	5	6	3	0	4	6	11
CHI	1	**55**	3	3	4	4	1	7	2	2	2
FRE	9	1	**40**	3	2	9	1	2	8	1	4
GER	3	7	4	**58**	1	5	0	0	1	1	5
HIN	0	1	0	0	**64**	1	2	0	0	14	1
ITA	7	1	5	3	4	**64**	1	0	5	0	4
JPN	3	15	2	0	2	4	**38**	12	8	1	0
KOR	2	21	1	2	2	2	9	**43**	4	1	3
SPA	6	8	8	2	5	9	7	8	**35**	3	9
TEL	2	1	0	2	34	1	0	0	2	**41**	0
TUR	9	2	0	5	3	6	2	1	3	1	**62**

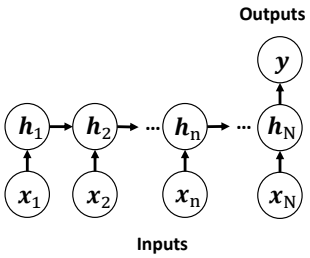

Figure 2: Many-to-one RNN structure used in the method of DNN with RNN(on sequence).

fusing the results on segments (see Equation 5) instead of fusing on speakers (see Equation 3).

$$P(j) = \frac{1}{S}\sum_{i=1}^{S}[w_{\text{DNN}}P_{\text{DNN}}(i,j) + w_{\text{RNN}}P_{\text{RNN}}(i,j)]. \quad (5)$$

Comparing the equation above with Equation (3), we see that the weights are inside the summation whereas they are outside the summation in (3).

For the fourth variation (DNN+ RNN (on sequence)), the structure is the same as that of the proposed system. The difference is in the way we train the RNN. In this method, we train the RNN on the segment level instead of on the frame level. In other words, the accent label was assigned to the segment instead of assigning it to every frame. This can be interpreted as a many-to-one model in Figure 2. The fusion between the DNN and the RNN was done in the same way as in the proposed system. The accuracy and UAR for these variations of the system are shown in Table 3. The results show that none of the variations of the system performs better than the current system.

Fusing with baseline. Comparing the results between the baseline and the proposed systems, we can see that the proposed system outperforms the baseline system overall and for most of the accents. However, for some of the accents, such as Spanish (SPA) and Telugu (TEL), the baseline system seems to work better than the proposed system. It seems that the neural networks and the SVM learned complementary representations of the data for the task. Therefore, we tried to fuse the prediction between the SVM-based baseline system and the proposed DNN/RNN based system. The weights of the fusion algorithm were tuned on the development set (set to 0.9 for the proposed system and 0.1 for the baseline system). The accuracy after fusing increased to 55.54%. The recalls are shown in the last column of Table 2. From the table, we can see the performance improved further after fusing with the baseline system. The confusion matrix is shown in Table 4.

Table 5: Confusion matrix on test set. Rows are reference, and columns are hypothesis.

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR
ARA	**28**	2	3	2	3	9	10	6	4	3	10
CHI	1	**45**	1	2	0	2	13	4	2	2	2
FRE	5	4	**38**	5	1	7	5	4	4	4	1
GER	0	7	6	**45**	1	4	1	0	3	1	7
HIN	5	3	1	2	**41**	0	0	0	2	27	1
ITA	5	3	7	2	2	**37**	0	0	6	4	2
JPN	5	5	0	2	1	1	**49**	10	0	0	2
KOR	2	13	1	1	1	1	12	**41**	4	0	4
SPA	7	5	10	4	4	8	5	4	**26**	1	3
TEL	1	1	0	0	29	0	0	2	0	**54**	1
TUR	14	4	5	2	1	2	4	2	4	1	**51**

Table 6: Recall for each class and the UAR on the test set (%)

ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	**UAR**
35	61	49	60	50	54	65	51	34	61	57	**52.5**

The best performance we can achieve on the test set is shown as confusion matrix in Table 5 and recalls in Table 6. The overall accuracy is 52.48% and the UAR is 52.48%. This is better than the performance of the baseline system reported in [23].

5. Discussion and Conclusion

In this paper, we present an accent identification system by combining DNNs and RNNs trained on long-term and short-term features respectively. We process the original speech samples into multiple segments to generate predictions of the accent type from each sample using neural networks, then to fuse them across all samples from a single speaker. Moreover, by fusing the results between DNNs and RNNs, we take advantage of both long-term prosodic features and short-term articulation features. We have evaluated the proposed system on the development set and the test set. The results show that the proposed system surpasses the performance of the provided SVM-based baseline system. By fusing the results of the proposed system with that of the baseline system, performance can be further improved. However, by looking through the confusion matrix in Table 4, we see that the system makes more mistakes among languages which are geographically close, such as between Hindi and Telugu; and among Japanese, Korean and Chinese. As future work it makes sense to develop a hierarchical classifier that initially considers groups of languages then makes more fine-grained decisions. Moreover, it is also worthwhile to investigate the individual benefits of DNNs and RNNs, since for some languages like Hindi, the prosody is more distinct; while for others like German, articulation is more important.

6. Acknowledgement

This work was partially supported by an NIH 1R21DC013812 grant. The authors graciously acknowledge a hardware donation from NVIDIA.

7. References

[1] L. Kat and P. Fung, "Fast accent identification and accented speech recognition," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, vol. 1. Phoenix, AZ, USA: IEEE, 1999, pp. 221–224.

[2] C. Huang, T. Chen, and E. Chang, "Accent issues in large vocabulary continuous speech recognition," *International Journal of Speech Technology*, vol. 7, no. 2-3, pp. 141–153, 2004.

[3] D. C. Tanner and M. E. Tanner, *Forensic aspects of speech patterns: voice prints, speaker profiling, lie and intoxication detection*. Lawyers & Judges Publishing Company, 2004.

[4] F. Biadsy, J. B. Hirschberg, and D. P. Ellis, "Dialect and accent recognition using phonetic-segmentation supervectors," 2011.

[5] L. M. Arslan and J. H. Hansen, "Frequency characteristics of foreign accented speech," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, vol. 2. Munich, Germany: IEEE, 1997, pp. 1123–1126.

[6] E. Ferragne and F. Pellegrino, "Formant frequencies of vowels in 13 accents of the british isles," *Journal of the International Phonetic Association*, vol. 40, no. 01, pp. 1–34, 2010.

[7] S. Deshpande, S. Chikkerur, and V. Govindaraju, "Accent classification in speech," in *Automatic Identification Advanced Technologies, Fourth IEEE Workshop on*. Buffalo, NY, USA: IEEE, 2005, pp. 139–143.

[8] P.-J. Ghesquiere and D. Van Compernolle, "Flemish accent identification based on formant and duration features," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, vol. 1. Orlando, FL, USA: IEEE, 2002, pp. I–749.

[9] 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 *Interspeech*. Lisbon, Portugal: Citeseer, 2005, pp. 217–220.

[10] T. Chen, C. Huang, E. Chang, and J. Wang, "Automatic accent identification using gaussian mixture models," in *Automatic Speech Recognition and Understanding, IEEE Workshop on*. Madonna di Campiglio, Italy: IEEE, 2001, pp. 343–346.

[11] H. Tang and A. A. Ghorbani, "Accent classification using support vector machine and hidden markov model," in *Advances in Artificial Intelligence*. Springer, 2003, pp. 629–631.

[12] K. Kumpf and R. W. King, "Foreign speaker accent classification using phoneme-dependent accent discrimination models and comparisons with human perception benchmarks," *in Proc. EuroSpeech*, vol. 4, pp. 2323–2326, 1997.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[14] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*. Brisbane, Australia: IEEE, 2015, pp. 4470–4474.

[15] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *Signal Processing Letters, IEEE*, vol. 21, no. 1, pp. 65–68, 2014.

[16] Y. Jiao, M. Tu, V. Berisha, and J. Liss, "Online speaking rate estimation using recurrent neural netwroks," in *Acoustics, Speech and Signal Processing, IEEE International Conference on*. Shanghai, China: IEEE, 2016.

[17] M. V. Chan, X. Feng, J. A. Heinen, and R. J. Niederjohn, "Classification of speech accents with neural networks," in *Neural Networks, IEEE World Congress on Computational Intelligence., IEEE International Conference on*, vol. 7. IEEE, 1994, pp. 4483–4486.

[18] A. Rabiee and S. Setayeshi, "Persian accents identification using an adaptive neural network," in *Second International Workshop on Education Technology and Computer Science*. Wuhan, China: IEEE, 2010, pp. 7–10.

[19] G. Montavon, "Deep learning for spoken language identification," in *NIPS Workshop on deep learning for speech recognition and related applications*, Whistler, BC, Canada, 2009, pp. 1–4.

[20] R. A. Cole, J. W. Inouye, Y. K. Muthusamy, and M. Gopalakrishnan, "Language identification with neural networks: a feasibility study," in *Communications, Computers and Signal Processing, IEEE Pacific Rim Conference on*. IEEE, 1989, pp. 525–529.

[21] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*. Florence, Italy: IEEE, 2014, pp. 5337–5341.

[22] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using long short-term memory recurrent neural networks." in *Interspeech*, Singapore, 2014, pp. 2155–2159.

[23] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, "The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language." in *Interspeech*. San Francisco, CA, USA: ISCA, 2016.

[24] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 1459–1462.

[25] M. Tu, X. Xie, and Y. Jiao, "Towards improving statistical model based voice activity detection." in *Interspeech*, Singapore, 2014, pp. 1549–1552.

[26] T. Virtanen, R. Singh, and B. Raj, *Techniques for noise robustness in automatic speech recognition*. John Wiley & Sons, 2012.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*. Vancouver, BC, Canada: IEEE, 2013, pp. 6645–6649.

[29] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[30] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.

[31] F. Chollet, "keras," https://github.com/fchollet/keras, 2015.