LEVERAGING NATIVE LANGUAGE SPEECH FOR ACCENT IDENTIFICATION USING DEEP SIAMESE NETWORKS

Aditya Siddhant[†]*, Preethi Jyothi*[§]*, Sriram Ganapathy*[‡]

[†]Carnegie Mellon University, Pittsburgh, USA
[§]Indian Institute of Technology Bombay, Mumbai, India
[‡]Indian Institute of Science, Bengaluru, India

ABSTRACT

The problem of automatic accent identification is important for several applications like speaker profiling and recognition as well as for improving speech recognition systems. The accented nature of speech can be primarily attributed to the influence of the speaker's native language on the given speech recording. In this paper, we propose a novel accent identification system whose training exploits speech in native languages as well as accented speech in English. Specifically, we develop a deep Siamese network based model which learns the association between accented speech recordings and the native language speech recordings. The Siamese networks are trained with i-vector features extracted from the speech recordings using both an unsupervised Gaussian mixture model (GMM) and a supervised deep neural network (DNN) model. We perform several accent identification experiments using the CSLU Foreign Accented English (FAE) corpus. In these experiments, our proposed approach using deep Siamese networks yield significant relative performance improvements of 15.4% on a 10-class accent identification task, over a baseline DNN-based classification system that uses GMM i-vectors. Furthermore, we present a detailed error analysis of the proposed accent identification system.

Index Terms— accent identification, i-vectors, deep Siamese networks

1. INTRODUCTION

Today, many voice-driven technologies are considered robust enough for daily use. This is largely due to significant advances in automatic speech recognition (ASR) technologies. However, variability in speech accents pose a significant challenge to state-of-the-art speech recognition systems. In particular, large sections of the English-speaking population in the world face difficulties interacting with voice-driven agents in English due to varying speech accents. The accented nature of speech can be primarily attributed to the influence of the speaker's native language. In this work we focus on the problem of *accent identification*, wherein a user's native language is automatically determined from their English speech. This can be viewed as a first step towards building accent-aware voice-driven systems.

Accent identification from non-native speech bears resemblance to the task of language identification [1]. However, accent identification is a harder task as many cues about the speaker's native language are lost or suppressed when the speech is in English. Nevertheless, one may expect that the speaker's native language is reflected in the acoustics of the individual phones used in English speech, along with pronunciations of words and grammar usage. In this work, we focus on the acoustic characteristics of an accent induced by a speaker's native language.

Our main contributions:

- We develop a novel deep Siamese network based model which learns the association between accented speech and native language speech.
- We explore i-vector features extracted using both an unsupervised Gaussian mixture model (GMM) and a supervised deep neural network (DNN) model.
- We present a detailed error analysis of the proposed system that reveals how accents produced by speakers from different linguistic backgrounds are similar or dissimilar to one another.

Section 3 outlines the i-vector feature extraction process. Section 4 describes our Siamese network-based model for accent identification. Our experimental results are detailed in Section 5 and Section 6 provides an error analysis of our system.

2. RELATED WORK

Prior work on foreign accent identification has drawn inspiration from techniques used in language identification [2]. Both phonotactic-based approaches [3] and acoustic-based approaches [4] have been explored for accent identification in the past. More recently, i-vector based representations, which led to state-of-the-art accuracies in speaker recognition [5]

and language recognition [6], have been applied to the task of accent recognition. i-vector systems that used GMM-based approaches were found to outperform other baseline systems [7, 8, 9].

All the previous i-vector-based approaches for accent identification used unsupervised GMMs to construct the i-vectors. They did not investigate the benefits of using supervised approaches to generate the sufficient statistics used in i-vector computation which would make the i-vectors more phonetically aligned. Also, none of the previous approaches exploited speech in native languages while training accent identification systems. This work address both these lines of enquiry.

3. FACTOR ANALYSIS FRAMEWORK

The techniques outlined here are derived from the previous work on joint factor analysis (JFA) and i-vectors [10, 11, 12]. We follow the notations used in [10]. The training data from all the speakers is used to train a GMM with model parameters $\lambda = \{\pi_c, \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}\}$ where π_c, $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ denote the mixture component weights, mean vectors and covariance matrices respectively for $c = 1, .., C$ mixture components. Here, $\boldsymbol{\mu}_c$ is a vector of dimension F and $\boldsymbol{\Sigma}_c$ is of assumed to be diagonal matrix of dimension $F \times F$.

3.1. I-vector Representations

Let $\boldsymbol{\mathcal{M}_0}$ denote the UBM supervector which is the concatenation of $\boldsymbol{\mu}_c$ for $c = 1, .., C$ and is of dimension of $CF \times 1$. Let $\boldsymbol{\Sigma}$ denote the block diagonal matrix of size $CF \times CF$ whose diagonal blocks are $\boldsymbol{\Sigma}_c$. Let $\mathcal{X}(s) = \{\boldsymbol{x}_i^s, i = 1, ..., H(s)\}$ denote the low-level feature sequence for input recording s where i denotes the frame index. Here $H(s)$ denotes the number of frames in the recording. Each $\boldsymbol{x}_i^s$ is of dimension $F \times 1$.

Let $\boldsymbol{\mathcal{M}(s)}$ denote the recording supervector which is the concatenation of speaker adapted GMM means $\boldsymbol{\mu}_c(s)$ for $c = 1, .., C$ for the speaker s. Then, the i-vector model is,

$$\boldsymbol{\mathcal{M}}(s) = \boldsymbol{\mathcal{M}_0} + \boldsymbol{V}\boldsymbol{y}(s) \tag{1}$$

where $\boldsymbol{V}$ denotes the total variability matrix of dimension $CF \times M$ and $\boldsymbol{y}(s)$ denotes the i-vector of dimension M. The i-vector is assumed to be distributed as $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

In order to estimate the i-vectors, the iterative EM algorithm is used. We begin with random initialization for the total variability matrix $\boldsymbol{V}$. Let $p_\lambda(c|\boldsymbol{x}_i^s)$ denote the alignment probability of assigning the feature vector $\boldsymbol{x}_i^s$ to mixture component c. The sufficient statistics are then computed as,

$$\begin{aligned} N_c(s) &= \sum_{i=1}^{H(s)} p_\lambda(c|\boldsymbol{x}_i^s) \\ \boldsymbol{S}_{X,c}(s) &= \sum_{i=1}^{H(s)} p_\lambda(c|\boldsymbol{x}_i^s)(\boldsymbol{x}_i^s - \boldsymbol{\mu}_c) \end{aligned} \tag{2}$$

Let $\boldsymbol{N(s)}$ denote the $CF \times CF$ block diagonal matrix with diagonal blocks $N_1(s)\boldsymbol{I}$, $N_2(s)\boldsymbol{I}$,..,$N_C(s)\boldsymbol{I}$ where $\boldsymbol{I}$ is the $F \times F$ identity matrix. Let $\boldsymbol{S}_X(s)$ denote the $CF \times 1$ vector obtained by splicing $\boldsymbol{S}_{X,1}(s)$,..,$\boldsymbol{S}_{X,C}(s)$.

It can be easily shown [10] that the posterior distribution of the i-vector $p_\lambda(\boldsymbol{y}(s)|\mathcal{X}(s))$ is Gaussian with covariance $\boldsymbol{l}^{-1}(s)$ and mean $\boldsymbol{l}^{-1}(s)\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s)$, where

$$\boldsymbol{l}(s) = \boldsymbol{I} + \boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{N}(s)\boldsymbol{V} \tag{3}$$

The optimal estimate for the i-vector $\boldsymbol{y}(s)$ obtained as $argmax_y\big[p_\lambda(\boldsymbol{y}(s)|\mathcal{X}(s))\big]$ is given by the mean of the posterior distribution.

For re-estimating the $\boldsymbol{V}$ matrix, the maximization of the expected value of the log-likelihood function (EM algorithm), gives the following relation [10],

$$\sum_{s=1}^{S} \boldsymbol{N}(s)\ \boldsymbol{V}\ \mathrm{E}\big[\boldsymbol{y}(s)\boldsymbol{y}^*(s)\big] = \sum_{s=1}^{S} \boldsymbol{S}_X(s)\mathrm{E}\big[\boldsymbol{y}^*(s)\big] \tag{4}$$

where E[.] denotes the posterior expectation operator. The solution for Eq. (4) can be computed for each row of $\boldsymbol{V}$. Thus, the i-vector estimation is performed by iterating between the estimation of posterior distribution and the update of the total variability matrix (Eq. (4)).

3.2. DNN i-vectors

Instead of using a GMM-UBM based computation of i-vectors, we can also use DNN context dependent state (senone) posteriors to generate the sufficient statistics used in the i-vector computation [13, 14]. The GMM mixture components will be replaced with the senone classes present at the output of the DNN. Specifically, $p_\lambda(c|\boldsymbol{x}_i^s)$ used in Eq. (2) is replaced with the DNN posterior probability estimate of the senone c given the input acoustic feature vector $\boldsymbol{x}_i^s$ and the number of senones is the parameter C. The other parameters of the UBM model $\lambda = \{\pi_c, \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}\}$ are computed as

$$\begin{aligned} \pi_c &= \frac{\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)}{\sum_{c=1}^{C}\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)} \\ \boldsymbol{\mu}_c &= \frac{\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)\boldsymbol{x}_i^s}{\sum_{c=1}^{C}\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)} \\ \boldsymbol{\Sigma}_c &= \frac{\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)(\boldsymbol{x}_i^s - \boldsymbol{\mu}_c)(\boldsymbol{x}_i^s - \boldsymbol{\mu}_c)^*}{\sum_{c=1}^{C}\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)} \end{aligned} \tag{5}$$

Using these estimates for the UBM parameters, the rest of the i-vector formulation discussed in Sec. 3.1 is followed to derive the DNN i-vectors. For the DNN i-vectors, we use a reduced set of senones (1088 obtained by merging the 10000 triphone states using a decision tree).

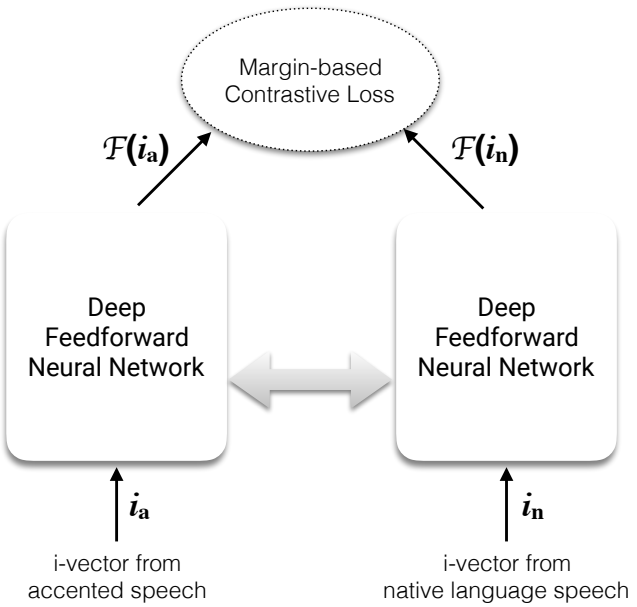

Fig. 1. Siamese network architecture for accent identification

4. OUR APPROACH

Siamese networks [15, 16] are neural network models that are designed to learn a similarity function between pairs of samples in an input space. This architecture consists of two identical neural networks with shared weights which take as input a pair of input samples. The objective is to find a function that minimizes or maximizes the similarity between the pair of inputs depending on whether or not they belong to the same category. This is achieved by optimizing a contrastive loss function containing dual terms to reduce the contribution from positive training examples (i.e. pairs of inputs belonging to the same category) and increase the contribution from negative training examples (i.e. pairs of inputs from different categories).

Figure 1 shows an illustration of the Siamese network we used for accent identification. Each training example comprises a pair of input i-vectors, $\{\mathbf{i}_1, \mathbf{i}_2\}$ corresponding to an accented English speech sample and a language speech sample, and a binary label $y \in \{0, 1\}$ indicating whether or not the native language underlying the accented English speech sample exactly matches the language speech sample. Positive training examples correspond to accented English i-vectors along with matched native language i-vectors from the same speaker. For the negative examples, we paired up accented English i-vectors with i-vectors from languages different from the one underlying the accented speech sample. These training instances are fed to twin networks with shared parameters which produce two outputs $\{\mathcal{F}(\mathbf{i}_1), \mathcal{F}(\mathbf{i}_2)\}$ corresponding to the input i-vectors $\mathbf{i}_1$ and $\mathbf{i}_2$. The whole network is then trained to minimize the following contrastive loss function:

$$\mathcal{L}(\mathbf{i}_1, \mathbf{i}_2, y) = (1 - y) \cdot d(\mathcal{F}(\mathbf{i}_1), \mathcal{F}(\mathbf{i}_2)) + y \cdot \max(0, 1 - d(\mathcal{F}(\mathbf{i}_1), \mathcal{F}(\mathbf{i}_2)) \quad (6)$$

where $d(\cdot, \cdot)$ is a distance function between representations.

We use a large number of training samples to learn a distance metric between the accented speech i-vectors and the language i-vectors. During test time, we compare the accented speech test i-vector with a representative language i-vector and choose the language whose feature representation is least distant from the accented speech i-vector according to the distance metric learned by the Siamese network. We experiment with different strategies to determine how the language i-vectors should be constructed during test time. Section 5 discusses more details of these test strategies.

5. EXPERIMENTAL RESULTS

5.1. Task Details

For our experiments, we used the CSLU Foreign Accented English Release 1.2 database [17] that consists of telephone-quality spontaneous English speech by native speakers of 22 different languages. We set up a 10-class accent identification task using accented English from speakers of 10 different languages which had the most data: Brazilian Portugese (BP), Hindi (HI), Farsi (FA), German (GE), Hungarian (HU), Italian (IT), Mandarin Chinese (MA), Russian (RU), Spanish (SP) and Tamil (TA). For native language speech, we used the CSLU 22 Languages Corpus [18] which contains telephone-quality continuous speech in all the above-mentioned 10 languages. Many of the speakers in the CSLU 22 Languages corpus also recorded speech samples for the CSLU Foreign Accented English corpus. (Samples from these speakers were used to construct positive examples for training our Siamese network.) Table 1 gives detailed statistics about the data used in our experiments, along with the training, development and test set splits.

Performance evaluation: We used accent identification accuracy as the metric to evaluate our proposed approach. This is computed as the percentage of utterances which are correctly identified as having one of the 10 above-mentioned accents. (A classifier based on chance would give an accuracy of 10% on this task.)

5.2. Comparing GMM i-vectors with DNN i-vectors

Table 2 shows the performance of various classifiers using both GMM i-vectors and DNN i-vectors as input features. LDA refers to a Linear Discriminant Analysis (LDA)-based classifier which performs supervised dimensionality reduction and reduces the dimensionality of the input vectors by linear projection onto a 9 dimensional space that maximizes

LANGUAGE	Accented speech			Native language speech
	Training	Dev	Test	
BP				
HI				
FA				
GE				
HU				
IT				
MA				
RU				
SP				
TA				

Table 1. Statistics of accented English and native language speech data. All the displayed numbers correspond to minutes of speech.

Classifier	GMM i-vectors		DNN i-vectors	
	Dev	Test	Dev	Test
LDA	-	37.2	-	43.8
SVM	-	40.2	-	45.2
NNET	-	40.8	-	44.8

Table 2. Accuracy rates from classifiers using both GMM i-vectors and DNN i-vectors.

the separation between classes. SVM refers to a SVM classifier using a radial basis function (RBF) kernel. NNET is a 3-layer feedforward neural network trained using the Adam optimization algorithm [19]. LDA and SVM were implemented using the mass package in R and NNET was implemented using Keras. Add more details about all three classifiers and citations for mass in R and Keras. We observe that the DNN i-vectors clearly outperform the GMM i-vectors. This is intuitive because the DNN i-vectors carry more information about the underlying phones. Both the SVM and NNET classifiers perform comparably.

5.3. Evaluating the Siamese network

All details about the Siamese network implementation. number of layers, distance function, initialization, number of hidden nodes, etc.

Table 4 compares the performance of our Siamese network to the best-performing baseline system (on the development set) from Table 2. We see consistent improvements from using the Siamese network classifiers over the best baseline system. Siamese-1 and Siamese-2 refer to two different test strategies that we used for the language i-vectors during test time. For the system labeled Siamese-1, we computed a mean language i-vector across all the i-vectors for a particular language. Add exact details about Siamese-2 i.e. language mean output + NNET classifier. We also tried generating a random sample of language i-vectors and computing the mean of the lowest five output scores, thus mimicking a soft-min function. This strategy performed comparably to Siamese-1.

Classifier	GMM i-vectors		DNN i-vectors	
	Dev	Test	Dev	Test
NNET	-	40.8	-	44.8
Siamese-1	-	-	-	46.8
Siamese-2	-	**42.3**	-	**47.9**

Table 3. Performance of Siamese network-based classifier.

5.4. Comparison with other methods using native language i-vectors

We also compare our Siamese network-based approach with other techniques that exploit speech data from native languages during training. Analogous to "NNET", we train a N-layer feedforward neural network but with input features consisting of language i-vectors appended to the accent i-vectors (referred to as "NNET-append"). Add details about the second no interconnection system which I didn't fully follow. Call it "NNET-append-2". Finally, we also investigate a transfer learning based approach (referred to as "NNET-transfer"). We train a N-layer feedforward neural network using only language i-vectors to predict the underlying language. Then, we use the resulting weights from the hidden layers as an initialization for a neural network that uses accent i-vectors as inputs to predict the underlying accent. Table **??** compares these three systems against our Siamese network-based approach. We observe our proposed Siamese-network based system performs better than all the other three systems which also use native language i-vectors.

5.5. Accuracies for utterances with varying properties

Add details about accent judgement, what 1-4 means.

5.6. Fusion

Add fusion results here.

Classifier	GMM i-vectors		DNN i-vectors	
	Dev	Test	Dev	Test
NNET-append	-	-	-	41.1
NNET-append-2	-	-	-	44.8
NNET-transfer	-	-	-	45.3
Siamese-2	-	**42.3**	-	**47.9**

Table 4. Performance of Siamese network-based classifier.

Duration (in secs)	Percentage of samples	Accuracy
0-5	-	-
6-10	-	-
11-15	-	-
16-20	-	-

Table 5. Accuracies for utterances of varying durations.

Accent judgement (1-4)	Percentage of samples	Accuracy
1	-	-
2	-	-
3	-	-
4	-	-

Table 6. Accuracies for utterances of varying accent strengths.

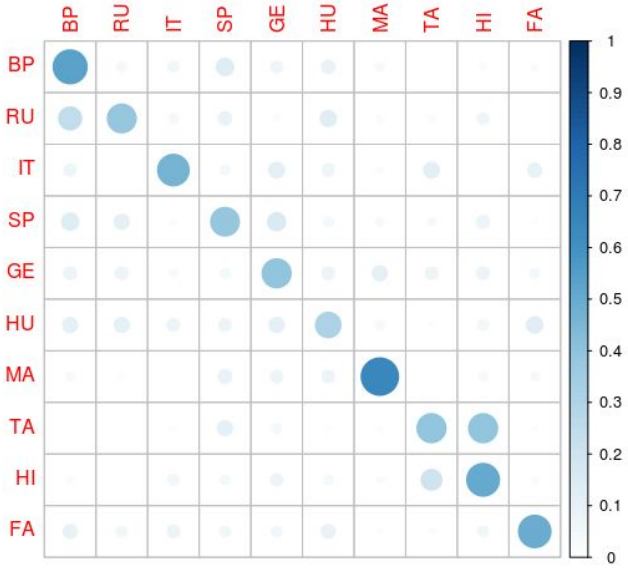

Fig. 2. Bubble plot visualizing the confusion matrix from the 10-class accent identification task

6. DISCUSSION

7. CONCLUSIONS

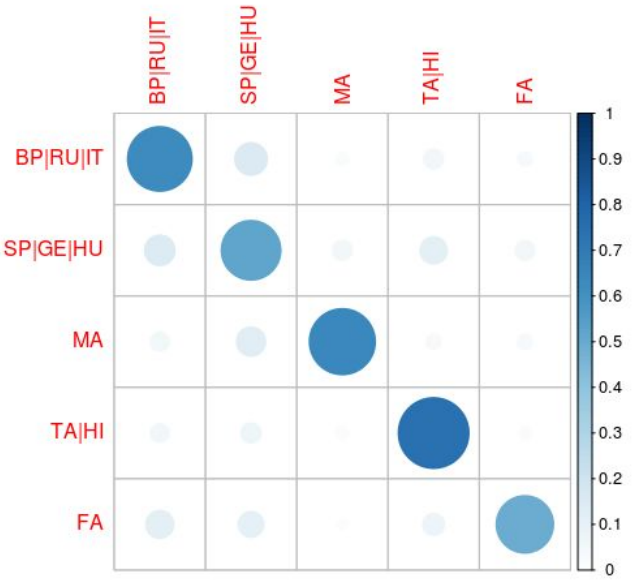

Fig. 3. Bubble plot visualizing the confusion matrix from the 10-class accent identification task, after grouping related languages.

8. REFERENCES

[1] Marc A Zissman and Kay M Berkling, "Automatic language identification," *Speech Communication*, vol. 35, no. 1, pp. 115–124, 2001.

[2] Marc A Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on speech and audio processing*, vol. 4, no. 1, pp. 31, 1996.

[3] F. Biadsy, J. Hirschberg, and D. P. W. Ellis, "Dialect and accent recognition using phonetic-segmentation supervectors," in *Proceedings of Interspeech*, 2011.

[4] Carlos Teixeira, Isabel Trancoso, and António Serralheiro, "Accent identification," in *Proceedings of ICSLP*. IEEE, 1996, pp. 1784–1787.

[5] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proceedings of Interspeech*, 2011.

[7] Mohamad Hasan Bahari, Rahim Saeidi, Hugo Van hamme, and David Van Leeuwen, "Accent recognition using i-vector, gaussian mean supervector

and gaussian posterior probability supervector for spontaneous telephone speech," in *Proceedings of ICASSP*. IEEE, 2013, pp. 7344–7348.

[8] Alexandros Lazaridis, Elie Khoury, Jean-Philippe Goldman, Mathieu Avanzi, Sébastien Marcel, and Philip N Garner, "Swiss french regional accent identification," in *Proceedings of Odyssey*, 2014.

[9] Maryam Najafian, Saeid Safavi, Phil Weber, and Martin Russell, "Identification of british english regional accents using fusion of i-vector and multi-accent phonotactic systems," in *Proceedings of Odyssey*, 2016.

[10] Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[11] Patrick Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[12] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[13] Mitchell McLaren, Yun Lei, Nicolas Scheffer, and Luciana Ferrer, "Application of convolutional neural networks to speaker recognition in noisy conditions," in *Proc. of INTERSPEECH*, 2014.

[14] Yun Lei, Luciana Ferrer, Aaron Lawson, Mitchell McLaren, and Nicolas Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," in *Proc. Speaker Odyssey Workshop*, 2014.

[15] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah, "Signature verification using a" siamese" time delay neural network," in *Proceedings of NIPS*, 1994, pp. 737–744.

[16] Sumit Chopra, Raia Hadsell, and Yann LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of CVPR*. IEEE, 2005, vol. 1, pp. 539–546.

[17] T. Lander, "CSLU: Foreign Accented English Corpus Release 1.2," 2007.

[18] T. Lander, "CSLU: 22 Languages Corpus," 2005.

[19] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.