TIME DELAY DEEP NEURAL NETWORK-BASED UNIVERSAL BACKGROUND MODELS FOR SPEAKER RECOGNITION

David Snyder, Daniel Garcia-Romero, Daniel Povey

Center for Language and Speech Processing & Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA

david.ryan.snyder@gmail.com, dgromero@jhu.edu, dpovey@gmail.com

ABSTRACT

Recently, deep neural networks (DNN) have been incorporated into i-vector-based speaker recognition systems, where they have significantly improved state-of-the-art performance. In these systems, a DNN is used to collect sufficient statistics for i-vector extraction. In this study, the DNN is a recently developed time delay deep neural network (TDNN) that has achieved promising results in LVCSR tasks. We believe that the TDNN-based system achieves the best reported results on SRE10 and it obtains a 50% relative improvement over our GMM baseline in terms of equal error rate (EER). For some applications, the computational cost of a DNN is high. Therefore, we also investigate a lightweight alternative in which a supervised GMM is derived from the TDNN posteriors. This method maintains the speed of the traditional unsupervised-GMM, but achieves a 20% relative improvement in EER.

***Index Terms*—** speaker recognition, deep neural networks, time delay neural networks, i-vector

1. INTRODUCTION

Modern speaker recognition systems are based on i-vectors [1]. In this paradigm, a universal background model (UBM) is used to collect sufficient statistics for i-vector extraction, and a probabilistic linear discriminant analysis (PLDA) backend computes a similarity score between i-vectors [2, 3, 4, 5, 6, 7]. Until recently, the state-of-the-art UBM was based on GMMs.

Recent speaker recognition systems have improved performance by replacing the GMM with a DNN to collect sufficient statistics (SS) for i-vector extraction [8, 9]. Usually, this DNN is trained as the acoustic model in an automatic speech recognition (ASR) system and is repurposed for speaker recognition. The output layer of a DNN provides soft alignments for phonetic content, often tied triphone states (or senones). These DNN posteriors are used in conjunction with features extracted using a standard approach for speaker recognition, to create the sufficient statistics for i-vector extraction [1]. The advantage of the DNN over the GMM may be due to its ability to directly model phonetic content, rather than an arbitrary acoustic space [8, 9, 10]. In [9] it was found that improvements to DNNs in terms of ASR word error rate (WER) may translate into improvements in speaker recognition performance. Recently, recurrent neural networks (RNN) and TDNNs [11] have outperformed traditional DNNs for a variety of LVCSR tasks [12, 13, 14]. In particular, the multisplice TDNN [14] had an 11% WER on Switchboard, better than RNN systems on the same task. Our DNN is based on [14].

The DNN-based speaker recognition methods achieve excellent results, but the performance comes at the cost of increased computational complexity. During i-vector extraction, the role of the UBM is to produce frame-level posteriors. For a DNN, the computation is nontrivial. In a resource limited application that nonetheless requires realtime performance, the DNN-based system may be impractical. Ideally, a supervised-GMM could be created with the speed of the traditional GMM-based UBM but with heightened phonetic awareness. In [15] a GMM-based ASR acoustic model replaced the usual GMM-UBM to create a phonetically aware GMM, but the improvements were only consistent during model combination [15].

Usually, DNN-based speaker recognition systems employ a supervised-GMM derived from the DNN posteriors and speaker recognition features (often not the same as the DNN features) [8, 9, 10]. However, this GMM is not typically used to collect SS; it has a minor role during i-vector extractor training. Promoting this supervised-GMM to the role of the UBM was explored in [8], but it did not improve on their baseline. It was speculated that this is due to the GMM's limited ability to model phonetic information. However, that GMM was diagonal, which possibly reduced its modeling capacity. In this paper we reexamine the value of this supervised-GMM as a lightweight alternative to the DNN-based speaker recognition system, and find that it consistently outperforms the baseline.

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1232825. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

2. EXPERIMENTAL SETUP

2.1. Datasets

We evaluate our systems on the condition 5 extended task of SRE10 [16]. The test consists of conversational telephone speech in enrollment and test utterances. In total there are 416,119 trials, over 98% of which are nontarget comparisons.

The UBM and i-vector extractor training data consists of male and female utterances from SWB and NIST SREs prior to 2010. The SWB data contains 1,962 speakers and 20,905 utterances of SWB Cellular and SWB 2 Phases II and III. The SRE dataset consists of 3,805 speakers and 36,614 utterances. To create in-domain systems, the PLDA backends are trained only on the SRE data. About 1,800 hours of the english portion of Fisher [17] is used to train the TDNN.

2.2. DNN Recipe

The system is based on the multisplice time delay DNN described in [14]. This architecture is currently the recommended recipe in the Kaldi toolkit [18] for large-scale speech recognition. In the multisplice system, a narrow temporal context is provided to the first layer and increasingly wide contexts are available to the subsequent hidden layers. The result is that higher levels of the network are able to learn greater temporal relationships.

The features are 40 MFCCs without cepstral truncation and with a frame-length of 25ms. These features are equivalent to filterbanks, but are more compressible. Cepstral mean subtraction is performed over a window of 6 seconds.

The TDNN has six layers, and a splicing configuration similar to those described [14]. Suppose t is some frame. At the input layer (layer 0) frames $[t-2, t+2]$ are spliced together. At layers 1, 3, and 4 we splice together frames $\{t-2, t+1\}$, $\{t-3, t+3\}$, and $\{t-7, t+2\}$, respectively. In total, the DNN has a left-context of 13 and a right-context of 9. The hidden layers use the p-norm (where $p = 2$) activation function [19]. The hidden layers have an input dimension of 350 and an output dimension 3500. The softmax output layer computes posteriors for 5297 triphone states. No fMLLR or i-vectors are used for speaker adaptation.

2.3. GMM-UBM Baseline

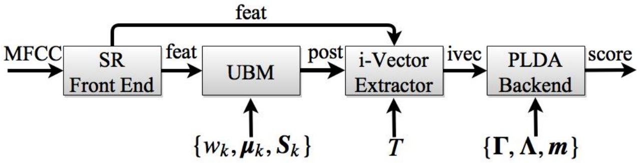
feat
MFCC
SR Front End
feat
UBM
post
i-Vector Extractor
ivec
PLDA Backend
score
$\{w_k, \boldsymbol{\mu}_k, \boldsymbol{S}_k\}$
$\boldsymbol{T}$
$\{\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{m}\}$

Fig. 1: GMM-based speaker recognition schema.

The UBM in our baseline system (illustrated in Figure 1) is a full-covariance GMM with several thousand mixture components. We compare systems with 2048, 4096, and 5297 components. The front-end consists of 20 MFCCs with a 25ms frame-length. The features are mean-normalized over a 3 second window. Delta and and acceleration are appended to create 60 dimensional frame-level feature vectors. The nonspeech frames are then eliminated using energy-based voice activity detection (VAD).

The GMM-UBM is trained on SWB and SRE datasets. It is initially trained for 4 iterations of EM using a diagonal covariance matrix and then for an additional 4 iterations with a full-covariance matrix. A 600 dimensional i-vector extractor is also trained on SWB and SRE for 5 iterations of EM. The backend consists of i-vector mean subtraction and length normalization, followed by PLDA scoring. To create an in-domain system, we estimate the i-vector mean $\boldsymbol{m}$ and the between-class and within-class covariance matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ of the PLDA backened using just the SRE dataset described in Section 2.1.

2.4. Supervised GMM-UBM

The goal of the supervised-GMM (shortened to sup-GMM) is to capture phonetic information useful to speaker recognition in a lightweight model. This is achieved by creating a GMM based on DNN posteriors and speaker recognition features. In contrast to the similar model in [8], our sup-GMM is full-covariance. The supervised and unsupervised GMMs differ only in the UBM training proceedure; during i-vector extraction, both systems follow the diagram in Figure 1.

We use the TDNN described in Section 2.2 to generate triphone posteriors on the SWB and SRE training data. Speaker recognition features (described in Section 2.3) are also computed on this training data. An energy-based VAD removes features and posteriors corresponding to nonspeech frames.

The mixture weights w_k, means $\boldsymbol{\mu}_k$ and covariances $\boldsymbol{S}_k$ are initialized according to Equation (1). The DNN parameters are collectively labeled $\boldsymbol{\Theta}$ and $Pr(k \mid \boldsymbol{y}_i, \boldsymbol{\Theta})$ is the probability of triphone k at frame i given the DNN features $\boldsymbol{y}_i$. The corresponding speaker recognition features are denoted $\boldsymbol{x}_i$.

$$\begin{aligned}
z_k^{(i)} &= Pr(k \mid \boldsymbol{y}_i, \boldsymbol{\Theta}), \\
w_k &= \sum_{i=1}^{N} z_k^{(i)}, \\
\boldsymbol{\mu}_k &= \frac{1}{w_k} \sum_{i=1}^{N} z_k^{(i)} \boldsymbol{x}_i, \\
\boldsymbol{S}_k &= \frac{1}{w_k} \sum_{i=1}^{N} z_k^{(i)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^{\top}.
\end{aligned} \tag{1}$$

Since the DNN output layer has 5297 senones, the sup-GMM also has 5297 components. Training of the $\boldsymbol{T}$ matrix and PLDA parameters $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ are unchanged from Section 2.3.

2.5. TDNN-UBM

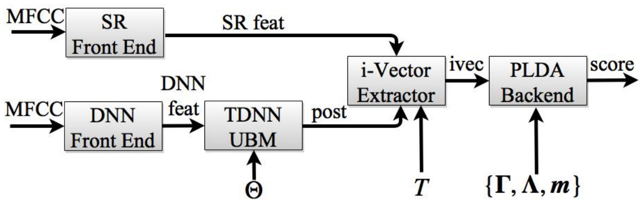

Fig. 2: TDNN-based speaker recognition schema.

This system uses the TDNN of Section 2.2 to create a UBM which directly models phonetic content. This system is based on the in-domain system described in [9] and is similar to those in [8] and [10]. The primary difference between this and earlier work is our utilization of the time delay DNN architecture. The supervised-GMM (Section 2.4) fills the role of the ancillary GMM of [9]. The parameters of the supervised-GMM are needed to initialize quantities required to train the i-vector extractor $\boldsymbol{T}$ matrix.

The only difference between this system and the preceding GMM-based system is the model used to compute frame-level posteriors. Here, TDNN posteriors and speaker recognition features are used for both i-vector extractor training and for computing enrollment and test i-vectors. The speaker recognition features are not the same as those used by the DNN, as illustrated by the two feature pipelines in Figure 2. To maintain the correct temporal context, an energy-based VAD is used to retain just the posteriors and features corresponding to speech frames. As in Sections 2.3 and 2.4, the i-vectors are 600 dimensional and the PLDA backend is trained just on the in-domain SRE data.

2.6. System Design

Experiments used ASR and speaker recognition modules in the Kaldi speech recognition toolkit [18]. Recipes for the systems described here are available in the SRE10 example of the Kaldi code repository (https://github.com/kaldi-asr/kaldi/tree/master/egs/sre10).

3. RESULTS

We compare gender independent and gender dependent versions of the baseline GMM, sup-GMM and TDNN systems. The gender independent systems each have a single pipeline which evaluates all of the SRE10 extended condition 5. The gender dependent systems share most of the same components with the gender independent systems. The SRE data is partitioned into male and female sets and two PLDA backends are trained. Accordingly, we evaluate the gender dependent models on just the male or female portions of SRE10. To avoid overly large tables we only report the performance for pooled gender dependent and independent scores. We evaluate recognition performance at three operating points: equal error-rate (EER) and normalized detection cost function (DCF) [16] with the probability of the target speaker set to 10^{-2} and 10^{-3}.

System	EER(%)	DCF10^{-3}	DCF10^{-2}
Sup-GMM-5297	1.94	0.388	0.213
TDNN-5297	1.20	0.216	0.123
GMM-2048	2.49	0.496	0.288
GMM-4096	2.56	0.468	0.287
GMM-5297	2.42	0.484	0.290

Table 1: Performance comparison of gender independent models on SRE10 C5.

System	EER(%)	DCF10^{-3}	DCF10^{-2}
Sup-GMM-5297	1.65	0.354	0.193
TDNN-5297	1.09	0.214	0.108
GMM-2048	2.16	0.417	0.239
GMM-4096	1.96	0.414	0.227
GMM-5297	2.00	0.410	0.241

Table 2: Performance comparison of gender dependent models on SRE10 C5.

In Tables 1 and 2 we see that there isn't much of a performance difference between the unsupervised GMMs with 2048, 4096 and 5297 components. We choose GMM-5297 as our primary baseline, since it has, by a small margin, the best gender independent EER of the baseline models.

Figures 3, 4, and 5 compare the performance among the GMM-5297, sup-GMM-5297 and TDNN-5297 systems. The DNN-based systems achieve the best results, with TDNN-5297 obtaining 1.20% and 1.09% gender independent and gender dependent EERs respectively. Figure 6 illustrates the relative improvement of the TDNN and sup-GMM over the GMM-5297 baseline. Across the three operating points with the gender independent and dependent systems we see a relative improvement of 13.65%-26.55% by the sup-GMM and 47.80%-57.59% by the TDNN. Although the performance of the sup-GMM isn't as good as the TDNN, it nevertheless outperforms the baseline by a significant margin. In similar methods such as [8] and [15] the supervised-GMM did not result in a significant improvement by itself. Perhaps the underlying reason lies in the high quality of the TDNN which the sup-GMM is based on. Additionally, full-covariance may allow the sup-GMM to retain modeling capacity.

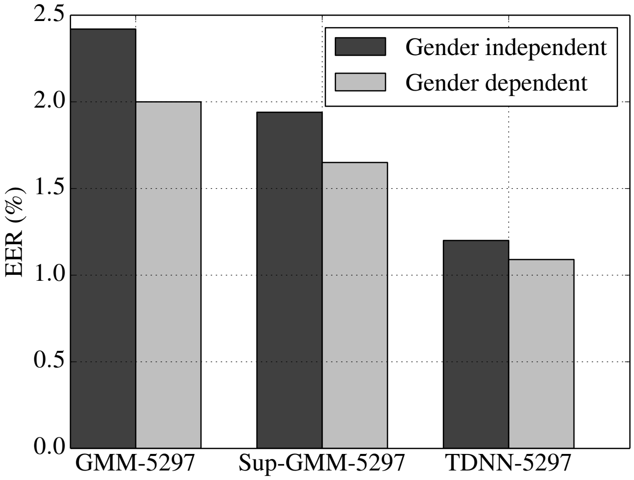

Fig. 3: Comparison of EERs.

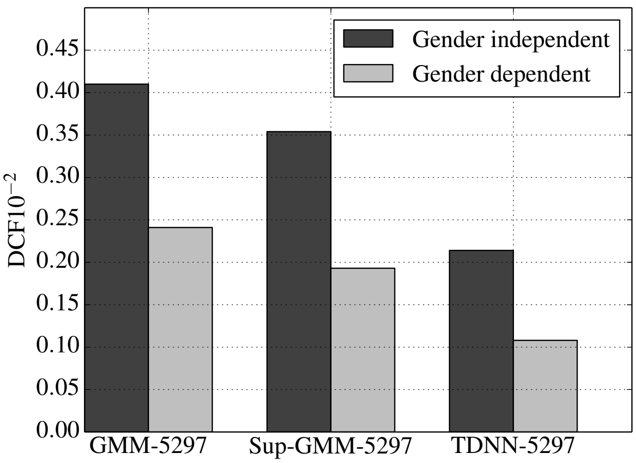

Fig. 4: Comparison of $DCF10^{-2}$.

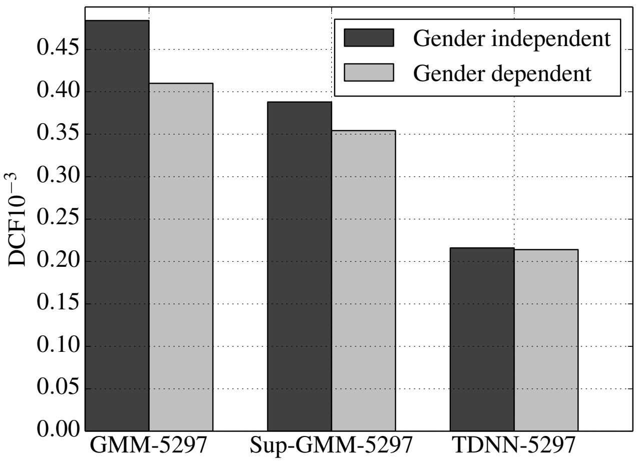

Fig. 5: Comparison of $DCF10^{-3}$.

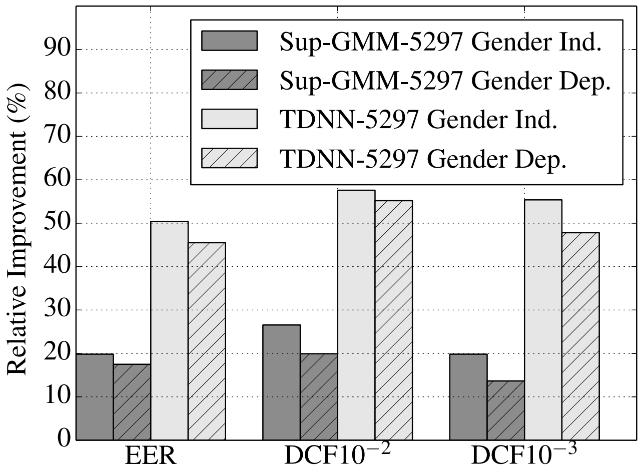

Fig. 6: Relative improvement over the GMM-5297 baseline.

The primary advantage of a GMM-based method lies in its efficiency during i-vector extraction. Using the sum of the `usr` and `sys` portions of the Linux tool `time` we recorded the duration of different parts of the system pipelines. In Table 3 and Figure 7, we represent this in terms of real-time factors. Ten 5 minute utterances were selected at random from the SRE10 test data and these were processed and timed from feature extraction to i-vector extraction 30 times. In Table 3 and Figure 7, i-vector extraction includes all computation needed to generate an i-vector after posteriors and features have been computed. The experiment was performed on an Intel x86-64 machine with 48 2000Mhz CPUs. The real-time factors were obtained by taking the average durations In CPU and dividing by the total utterance length.

The GMM-2048 system is about twice as fast as the larger GMMs with 4096 or 5297 components during posterior and i-vector extraction. Even without parallelization the GMM-based systems are at least ten times faster than real-time. Since the TDNN system needs to compute features for both the DNN and for speaker recognition, this stage of the pipeline is about twice as slow as the GMM-based systems. Without parallelization, the vast majority of the DNN-based system is spent in the posterior calculation. This results in a system which is nearly 36% slower than real-time, and more than ten times slower than the sup-GMM-5297.

In practice we would perform the DNN posterior matrix calculations in CUDA to obtain faster than real-time performance. However, by comparing the total CPU time between the systems, we expose the overall computational load of the DNN, and facilitate a comparison of compute-cost vs. performance of the three systems.

4. CONCLUSION

We explored the use of TDNNs for speaker recognition on the SRE10 task. We found that this DNN yields a large rel-

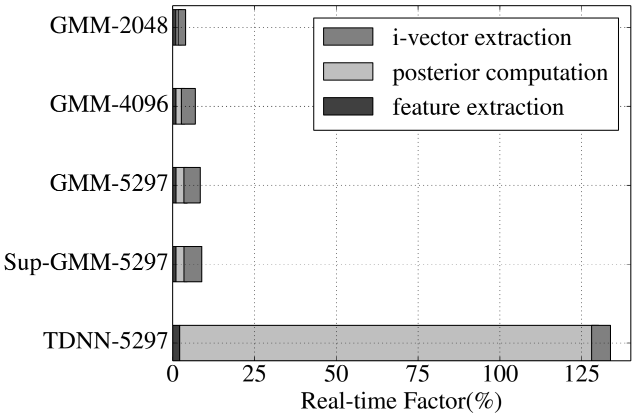

Fig. 7: Total CPU time relative to utterance length for each system.

Table 3: CPU time relative to utterance length for primary stages of the system pipelines.

System	Feat.(%)	Post.(%)	i-Vec.(%)
Sup-GMM-5297	1.02	3.46	5.44
TDNN-5297	2.15	128.02	5.77
GMM-5297	1.03	3.44	5.01
GMM-4096	1.01	2.67	4.27
GMM-2048	1.01	1.68	2.28

ative improvement over the unsupervised GMM baseline on EER and DCF operating points. With the TDNN-UBM we also achieve a 1.20% gender independent EER, which we believe is the best reported on the task. We also highlighted the computational advantages of the GMM over the DNN, and showed that there is a significant cost for computing DNN posteriors. While GPU parallelization is commonly used to obtain real-time performance, it may not be feasible for all applications. We found that the supervised-GMM, normally of minor use in the DNN-based system, can be used on its own as a fast alternative to the DNN with better performance than the baseline.

5. REFERENCES

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

[2] S.J.D. Prince and J.H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct 2007, pp. 1–8.

[3] Niko Brümmer and Edward De Villiers, "The speaker partitioning problem.," in *Odyssey*, 2010, p. 34.

[4] Patrick Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Odyssey*, 2010, p. 14.

[5] Jesús A Villalba and Niko Brümmer, "Towards fully Bayesian speaker recognition: Integrating out the between-speaker covariance.," in *INTERSPEECH*, 2011, pp. 505–508.

[6] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Interspeech*, 2011, pp. 249–252.

[7] Daniel Garcia-Romero, Xinhui Zhou, and Carol Y Espy-Wilson, "Multicondition training of Gaussian plda models in i-vector space for noise and reverberation robust speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4257–4260.

[8] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Moray McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[9] Daniel Garcia-Romero, Xiaohui Zhang, Alan McCree, and Daniel Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 378–383.

[10] Patrick Kenny, Vishwa Gupta, Themos Stafylakis, P Ouellet, and J Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Proc. Odyssey*, 2014.

[11] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, Mar 1989.

[12] Hasim Sak, Andrew W. Senior, and Françoise Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *CoRR*, vol. abs/1402.1128, 2014.

[13] George Saon, Hagen Soltau, Ahmad Emami, and Michael Picheny, "Unfolded recurrent neural networks for speech recognition," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[14] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," `http://www.danielpovey.com/files/2015_interspeech_multisplice.pdf`, 2015, to appear in the Proceedings of *INTERSPEECH*.

[15] M. Omar and J. Pelecanos, "Training universal background models for speaker recognition," in Odyssey: The Speaker and Language Recognition Workshop, 2010.

[16] "The NIST year 2010 speaker recognition evaluation plan," `http://www.itl.nist.gov/iad/mig/tests/sre/2010/`, 2010.

[17] Christopher Cieri, David Miller, and Kevin Walker, "The Fisher corpus: a resource for the next generations of speech-to-text.," in *LREC*, 2004, vol. 4, pp. 69–71.

[18] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," 2011.

[19] Xiaohui Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 215–219.