A Deep Neural Network Speaker Verification System Targeting Microphone Speech

Yun Lei[1] *Luciana Ferrer*[1,2] *Mitchell McLaren*[1] *Nicolas Scheffer*[1]

[1] Speech Technology and Research Laboratory, SRI International, California, USA
[2] Departamento de Computación, FCEN, Universidad de Buenos Aires and CONICET, Argentina
{yunlei,lferrer,mitch,scheffer}@speech.sri.com

Abstract

We recently proposed the use of deep neural networks (DNN) in place of Gaussian Mixture models (GMM) in the i-vector extraction process for speaker recognition. We have shown significant accuracy improvements on the 2012 NIST speaker recognition evaluation (SRE) telephone conditions. This paper explores how this framework can be effectively used on the microphone speech conditions of the 2012 NIST SRE. In this new framework, the verification performance greatly depends on the data used for training the DNN. We show that training the DNN using both telephone and microphone speech data can yield significant improvements. An in-depth analysis of the influence of telephone speech data on the microphone conditions is also shown for both the DNN and GMM systems. We conclude by showing that the the GMM system is always outperformed by the DNN system on the telephone-only and microphone-only conditions, and that the new DNN / i-vector framework can be successfully used providing a good match in the training data.

Index Terms: Deep neural networks, Microphone data, Speaker recognition, i-vectors

1. Introduction

The speaker verification community has seen a significant increase in accuracy from the successful application of the i-vector extraction paradigm [1]. This framework can be decomposed into three sequential stages: the collection of sufficient statistics, the extraction of i-vectors and a probabilistic linear discriminant analysis (PLDA) backend. The collection of sufficient statistics is a process where a sequence of feature vectors (e.g., mel-frequency cepstral coefficients (MFCC)) are represented by the Baum-Welch statistics obtained with respect to a GMM, referred to as a universal background model (UBM) [2]. These high dimensionality statistics are converted into a single low-dimensional feature vector — an i-vector — that represents important information about the speaker along with other types of variability in a given speech segment. Once i-vectors are extracted, a PLDA model is used to compensate for nuisance variability and produce verification scores by comparing i-vectors extracted from different utterances [3].

Recently, we proposed a new DNN/i-vector framework that uses a DNN trained for speech recognition [4] to guide speaker modelling, specifically, by using the output posteriors as frame alignments for speaker modelling and i-vector extraction [5]. In this approach, the DNN takes the place of the UBM in the standard framework and is used to compute the posterior of the frames with respect to each of the classes in the model. In the case of the UBM, the classes are the individual Gaussians from a mixture model; in the case of the DNN, the classes were senones (tied triphone states) obtained using a standard decision tree for automatic speech recognition. Once the posteriors are computed, the zero-th and first order statistics are computed in the standard way before they are fed into the state-of-the-art i-vector / PLDA paradigm. In [5], this new DNN/i-vector framework has shown to significantly outperform a series of UBM/i-vector baselines in the NIST SRE'12 telephone conditions.

In this work, we extend this framework to microphone speech and combined microphone/telephone conditions. Specifically, we measure the data influence in each stage of the DNN/i-vector pipeline, highlighting areas of data sensitivity and how these differ from the traditional UBM/i-vector training regimes. Finally, we compare the microphone-ready DNN/i-vector system to UBM/i-vector systems on a range of conditions from the NIST SRE'12 set that involve microphone channels.

2. The DNN/i-vector Framework

In the i-vector model [1], the t-th speech frame $\mathbf{x}_t^{(i)}$ from the i-th speech segment is assumed to be generated by the following distribution:

$$\mathbf{x}_t^{(i)} \sim \sum_k \gamma_{kt}^{(i)} \mathcal{N}(\boldsymbol{\mu}_k + \mathbf{T}_k \boldsymbol{\omega}^{(i)}, \boldsymbol{\Sigma}_k) \quad (1)$$

where the $\mathbf{T}_k$ matrices describe a low-rank subspace (called total variability subspace) by which the means of the Gaussians are adapted to a particular speech segment, $\boldsymbol{\omega}^{(i)}$ is a segment-specific standard normal-distributed latent vector, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance of the k-th Gaussian, and $\gamma_{kt}^{(i)}$ are the alignments of $\mathbf{x}_t^{(i)}$. In general, we represent the alignments by the posterior of the k-th Gaussian, given by:

$$\gamma_{kt}^{(i)} = p(k|x_t^{(i)}) \quad (2)$$

The i-vector used to represent the speech signal is the maximum *a posteriori* (MAP) point estimate of the latent vector $\boldsymbol{\omega}^{(i)}$.

Equation (1) models a process by which the frame for time t is generated by first choosing a class k according to the distribution given by Equation (2) and then generating the features according to the Gaussian distribution for that class, $\mathcal{N}(\boldsymbol{\mu}_k + \mathbf{T}_k \boldsymbol{\omega}^{(i)}, \boldsymbol{\Sigma}_k)$. Note that the classes can be defined in any way, subject to the theoretical restriction that the classes have a Gaussian distribution.

Given a speech segment, the following sufficient statistics can be computed using the posterior probabilities of the classes:

$$\begin{aligned} \mathbf{N}_k^{(i)} &= \sum_t \gamma_{kt}^{(i)} \\ \mathbf{F}_k^{(i)} &= \sum_t \gamma_{kt}^{(i)} x_t^{(i)} \end{aligned} \quad (3)$$

These statistics are used to train the subspace $\mathbf{T}$ and extract the i-vector $\boldsymbol{\omega}^{(i)}$. Note that while means and covariances in Equation (1) can be estimated during the subspace training process, they are usually pre-computed [6].

Traditionally, the Gaussians in a GMM-UBM define the classes k in Equation (1) and the posteriors for the classes are computed from the likelihoods of the Gaussians using the Bayes rule [1, 6]. The GMM-based approach ensures that the Gaussian approximation for each class is satisfied (by definition) and provides a relatively simple way to obtain the posteriors.

In our recent work [5], we proposed defining the classes k in Equation (1) as the senones determined by a decision tree trained for automatic speech recognition (ASR) instead of the Gaussian indices in a GMM. The senones are defined as states within context-dependent phones. They can be, for example, each of the three states within all triphones. They are the unit for which observation probabilities are computed during ASR. The pronunciations of all words are represented by a sequence of senones $\mathcal{Q}$. By using the set $\mathcal{Q}$ to define the classes k, we make the assumption that each of these senones can be accurately modelled by a single Gaussian. While this is a strong assumption, previous results showed that it is probably a reasonable one for the NIST SRE'12 clean telephone task, since results on this task using the proposed system greatly outperform the state of the art. To obtain the posteriors for each of the k classes, now defined as senones, a DNN is trained to predict each of the senones [7]. A softmax function is used in the last layer to generate the required posteriors.

In the traditional GMM/i-vector framework, the means and covariance matrices of the GMM-UBM are used as the means $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ of the classes defined in equation (1). In the DNN/i-vector framework, the means $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ of the senones are given by:

$$\begin{aligned} \gamma_{kt}^{(i)} &\approx p(k|x_t^{(i)}) \\ \boldsymbol{\mu}_k &= \frac{\sum_{i,t} \gamma_{kt}^{(i)} x_t^{(i)}}{\sum_{i,t} \gamma_{kt}^{(i)}}, \\ \boldsymbol{\Sigma}_k &= \frac{\sum_{i,t} \gamma_{kt}^{(i)} x_t^{(i)} x_t^{(i)^T}}{\sum_{i,t} \gamma_{kt}^{(i)}} - \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T. \end{aligned} \quad (4)$$

where the ASR system is used to compute the posteriors $p(k|x_t^{(i)})$ for each class k for each frame and $x_t^{(i)}$ are the acoustic features used for speaker recognition, which can be different from the features used in the DNN.

Figure 1 presents a flow diagram of the DNN/i-vector hybrid framework. ASR features (e.g., log-Mel filterbanks) are input to the DNN, which generates posteriors for each senone for each frame. These posteriors are used to generate zeroth- and first-order statistics, as well as means and covariances defined by equations 4, using features suited to speaker verification (SV) (e.g., MFCC appended with deltas). These statistics and the parameters (e.g., means and covariances) are used for the subsequent i-vector subspace training. Finally, the i-vectors are scored by the LDA/PLDA backend.

3. Experiment Protocol

Many different systems are tested in the following sections for a thorough analysis of the impact of training data on each stage of the DNN/i-vector framework in the context of microphone speech conditions. To reduce the computational burden, experiments are constrained to female trials and no second order

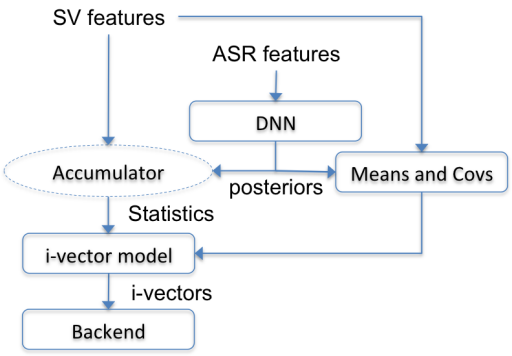

Figure 1: The flow diagram of the DNN/i-vector framework where the DNN is trained on the ASR data, and means and covariances (e.g., $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ in equation 1 and 4), i-vector model and backend are trained on the speaker verification data.

derivatives (double deltas) are used. It was confirmed that the exclusion of double deltas resulted in only a marginal loss of performance for both DNN and UBM frameworks.

For the baseline UBM/i-vector model, 20 MFCC coefficients (including C0), augmented with first order derivatives (deltas) are used. A 2048 diagonal component UBM is trained in a gender-dependent fashion, along with a 400 dimensional i-vector extractor on the same data set. The dimensionality of the i-vectors is further reduced to 200 by LDA, followed by length normalization and PLDA. Because the DNN system uses roughly 3500 classes, another system is trained with a 4096 diagonal component UBM for comparison. More details can be found in [8]

Approximately 3500 senones are obtained using decision trees as described in [9]. Standard HMM-GMM ASR systems are used to generate the initial senone alignments to train the DNNs. These cross-word triphone HMM-GMM ASR systems including roughly 200K Gaussians are trained with maximum likelihood (ML). The features used in the HMM-GMM model are 39-dimensional MFCC features, consisting of 13 static features (including C0) and first and second order derivatives. The features are pre-processed with speaker-based cepstral mean and covariance normalization (MVN). A seven-layer DNN with 1200 nodes in each hidden layer is trained with cross entropy using the alignments from the HMM-GMM system. The input layer of the DNN is composed of 15 frames (7 frames on each side of the frame for which predictions are made), where each frame corresponds to 40 log mel-filterbank coefficients. The DNN is used to provide the posterior probabilities for the senones at each frame. For consistency, the same features from the baseline system, MFCC+deltas, are used to compute sufficient statistics from the frame alignment given by the DNN and system hyperparameters are also matched to the baseline system.

System performance is reported in terms of detection cost function (DCF) with different effective priors, equal error rate (EER), and the false alarm rate at a miss rate of 10% (M10). The effective priors for the two DCFs are 0.001 and 0.01 as defined in NIST SRE'12 [10].

4. Results

In this section, we attempt to highlight the data sensitivity of the DNN/i-vector framework with respect to the training data on microphone speech conditions, and how this sensitivity might

contrast with the traditional UBM/i-vector framework. We first evaluate DNNs differing in their training data on the SRE'12 extended microphone condition using only microphone data for enrollment. We then present an analysis of the telephone data's influence on the model training for the microphone conditions using only microphone data for enrollment. Finally, we show results of the DNN/i-vector framework on the NIST SRE'12 extended telephone and microphone conditions [10] where both types of channels are used for enrollment.

4.1. Influence of Telephone Data for DNN Training on SRE'12 C1-mic Condition

In this section, we evaluate the DNN/i-vector framework on the extended NIST SRE'12 microphone condition C1 (no added noise in testing). To avoid any influence from the telephone data, we first focus on the condition in which only microphone speech data is used for enrollment. The condition consists of 5,508,514 trials, including 2,790 target and 5,505,724 impostor trials. We refer to this condition as C1-mic.

Two DNN models are used in the experiments: the DNN trained on microphone data only (DNN-mic), and the DNN trained on both microphone and telephone data (DNN-both). The DNN trained on telephone data is not considered here, since we only focus on the microphone conditions. For training the DNN, the microphone data includes meeting recordings from the AMI, CMU, ICSI, and NIST corpora. The total meeting time is about 200 hours after speech/nonspeech segmentation, while the amount of speech data actually used in the ASR model training is roughly 800 hours, including the speech from different microphones (e.g., distant and close-talking microphones) [11]. As in our previous work [5], the telephone data includes roughly 1300 hours of clean English telephone speech from the Fisher, Callhome, and Switchboard data sets. The HMM-GMM models used for alignment are trained on the same data used for the DNNs.

Table 1 presents the performance of the two baseline systems and the two DNN/i-vector systems on the C1-mic condition. To initially compare the system performance without any influence from the telephone data, system components (including UBM, i-vector subspaces, and the backends) are trained on microphone data only. There are 9,276 files used for UBM and i-vector subspace training, and 9,517 files with 382 speakers used for the PLDA backend training. Note that in the DNN-both case, only microphone speech data is used to compute the means and covariances in equation (4). This is consistent with the fact that only microphone data is used to train the UBM for the baseline systems.

Table 1: *The performance of the DNN/i-vector systems compared to the baseline systems on the NIST SRE'12 extended microphone condition with only microphone data for enrollment (C1-mic). The UBM, i-vector space and PLDA backend are trained using only microphone data. Two DNNs are compared: DNN-mic, trained with only microphone data, and DNN-both, trained with telephone and microphone data.*

System	$DCF_{0.001}$	$DCF_{0.01}$	EER(%)	M10
UBM(2048)	0.341	0.207	2.40	0.17
UBM(4096)	0.358	0.210	2.37	0.17
DNN-mic	0.339	0.210	2.79	0.21
DNN-both	0.309	0.181	2.15	0.12

The table shows that the DNN trained on only microphone data (DNN-mic) is comparable to both baseline systems in DCF measures but performs worse on the other measurements. By contrast, the DNN trained on both telephone and microphone data performs significantly better than both baseline systems across all measurements[1]. One hypothesis for the lack of improvement found with the DNN-mic system over the baseline systems is the mismatch between microphone data used for DNN training and the NIST SRE microphone data. The addition of the telephone data into the DNN training yields significant gains with respect to using only microphone data for training, similar to the findings in the ASR tasks [12]. This might also indicate that the telephone data is simply adding more diversity to the training data resulting in a DNN that generalizes better to unseen microphones. Based on these experiments, the DNN trained on both telephone and microphone data is used in the rest of the experiments.

4.2. Influence of Telephone Data for Model Training on SRE'12 C1-mic Condition

The previous section demonstrated how the DNN/i-vector framework can be successfully applied to microphone speech when only microphone data is used for training the components of the verification framework: UBM, i-vector subspace and PLDA backend. Traditionally, however, both microphone and telephone data are used when training these models. For this reason, we explore the performance of the system when adding telephone training data to each system component.

Table 2 presents the performance of different systems where the UBM for the baseline systems and the means and covariance for the DNN/i-vector approach (see equation(4)) are obtained on both telephone ($\sim$ 40k files) and microphone data, while the other parts of the pipeline (i-vector subspace and the backend) use microphone training data only. These results are presented on the C1-mic condition in Table 2.

Table 2: *The performance on the C1-mic condition where both telephone and microphone data are used to train the UBM in the baseline systems and the means and covariances from Equation (4). The i-vector subspace and the backend are trained on microphone data only.*

System	$DCF_{0.001}$	$DCF_{0.01}$	EER(%)	M10
UBM(2048)	0.347	0.214	2.51	0.20
UBM(4096)	0.364	0.222	2.54	0.22
DNN-both	0.301	0.178	2.11	0.11

The results in Table 2 are consistent with the conclusions obtained from Table 1, where the performance of the DNN-both system is still significantly better than that of the baseline systems. In addition, since the results in Tables 2 and 1 are quite similar, we conclude that the data used for estimating the UBM or the means and covariances defined in equation (4) does not significantly affect performance.

Next, we evaluate the influence of the data on the i-vector subspace model. Table 3 shows the system performance where i-vectors are generated using a UBM, means and covariances and i-vector subspace trained on telephone and microphone data, while the backend is trained on microphone data only.

Table 3 shows that the UBM/i-vector baseline systems improve from this change compared to the results in Table 2,. Furthermore, the UBM with 4096 Gaussians outperforms the UBM

[1]Note that the fact that the UBM with 4096 Gaussians performs slightly worse than the UBM with 2048 Gaussians in this table might be caused by the relatively small amount of data in the i-vector model training compared to that available when including telephone data.

Table 3: *Performance comparison on the C1-mic condition where both telephone and microphone data are used to train the UBM in the baseline systems and the means and covariances defined in equation (4) as well as the i-vector subspace. The backend is trained on microphone data only.*

System	$DCF_{0.001}$	$DCF_{0.01}$	EER(%)	M10
UBM(2048)	0.312	0.201	2.62	0.19
UBM(4096)	0.298	0.185	2.26	0.13
DNN-both	0.298	0.195	2.83	0.20

with 2048 Gaussians due to the much larger training set used in the i-vector subspace training. Despite this increase in training data, the DNN/i-vector framework suffers a significant performance degradation. We conclude that the telephone data used for the i-vector subspace training affects the UBM/i-vector and DNN/i-vector systems differently. We believe this difference might be caused by the different definition of the classes used in the i-vector model. For example, the UBM defines the classes based on the data in an unsupervised way while the DNN defined the classes based on the senones. The data distribution of each senone in the mixed case might be far from the Gaussian distribution assumed by the i-vector model.

Finally, Table 4 shows the system performance where all the models are trained on both telephone and microphone data, Comparing to Table 3, the performances of all systems here are slightly worse due to adding mismatched data into PLDA training. However, as in Table 3, the DNN/i-vector system performs similar to the baseline systems, losing the advantage observed in Table 1.

Table 4: *Performance comparison on the C1-mic condition where all models are trained on both telephone and microphone data.*

System	$DCF_{0.001}$	$DCF_{0.01}$	EER(%)	M10
UBM(2048)	0.348	0.212	2.23	0.17
UBM(4096)	0.333	0.194	1.97	0.11
DNN-both	0.334	0.194	2.58	0.14

4.3. Performance on SRE'12 Extended Conditions

We now extend our analysis to other conditions, and Table 5 presents the overall performance of the DNN/i-vector system compared to the baseline systems, on three extended NIST SRE'12 conditions: microphone speech (C1), telephone speech (C2), and telephone speech collected under noisy conditions (C5). All enrollment data is used which includes both telephone and microphone recordings. Since the DNN used here is trained on clean data only, conditions C3 and C4 which have added noise are not considered here. Condition C5 was included however as the level of noise is known to be significantly lower than C4. All systems in this table are trained on both telephone and microphone data.

We can see that similar to [5], the impressive gains from the DNN/i-vector framework on the two telephone conditions are still observed even though the mis-matched microphone data was added into the DNN/i-vector system and speaker models. In the microphone condition, the DNN/i-vector system performs similarly to the UBM/i-vector systems on both DCF metrics and M10. Notably, a slight degradation was observed on EER. Based on the analysis performed in the previous section, this degradation might be caused by the mismatch between the data used for DNN training, for verification and for the i-vector subspace training on both channels.

Table 5: *Performance comparison on the NIST SRE'12 extended C1, C2, and C5 conditions.*

a. The performance on $DCF_{0.001}$.

System	C1	C2	C5
UBM(2048)	0.273	0.366	0.417
UBM(4096)	0.252	0.341	0.391
DNN-both	0.262	0.280	0.304

b. The performance on $DCF_{0.01}$.

System	C1	C2	C5
UBM(2048)	0.155	0.215	0.268
UBM(4096)	0.137	0.199	0.250
DNN-both	0.150	0.147	0.175

c. The performance on EER (%).

System	C1	C2	C5
UBM(2048)	1.80	2.19	3.19
UBM(4096)	1.55	2.01	2.76
DNN-both	2.01	1.40	1.87

d. The performance on M10.

System	C1	C2	C5
UBM(2048)	0.06	0.18	0.39
UBM(4096)	0.04	0.13	0.29
DNN-both	0.05	0.05	0.09

5. Conclusions and Future Work

In previous work [5], we presented the DNN/i-vector framework for speaker recognition and showed its performance on the SRE'12 telephone conditions using only telephone data for training all system components. In this work, we extended its application to microphone speech data. We found that training the ASR DNN with both telephone and microphone data provided better verification performance on microphone test data than training on only microphone data. This result may indicate that the microphone data used for this purpose is not well matched to the test data and that the additional telephone data improves model robustness by increasing diversity.

With regard to the influence of data on the system's component (UBM, means and covariances, i-vector subspace and backend), it was found that the i-vector subspace was sensitive to the addition of telephone training data to the microphone training set. In the case of the traditional UBM framework, this significantly improved microphone trial performance while, in contrast, the DNN/i-vector framework showed a significant degradation.

Finally, we reported the performance of the DNN/i-vector framework on both telephone and microphone SRE'12 conditions and show that gains are obtained on telephone conditions, similar to [5], when the system is trained with both telephone and microphone data, while microphone results become comparable to those of the baseline systems.

The results shown in this work highlight two major directions for further work on the DNN/i-vector framework: (1) the study of strategies for dealing with data mismatch issues between the DNN and verification framework training; and (2) the optimization of the i-vector subspace modelling when faced with diverse data.

6. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. ASLP*, vol. 19, pp. 788–798, May 2010.

[2] D. A. Reynolds, T. F. Quatieri, and D. R. B., "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19 – 41, 2000.

[3] S. Prince, "Probabilistic linear discriminant analysis for inferences about identity," in *ICCV-2007*. IEEE, 2007, pp. 1–8.

[4] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "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.

[5] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *ICASSP-2014*. IEEE, 2007.

[6] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. ASLP*, vol. 16, pp. 980–988, July 2008.

[7] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. ASLP*, vol. 20, pp. 30–42, 2012.

[8] L. Ferrer, M. McLaren, N. Scheffer, Y. Lei, M. Graciarena, and V. Mitra, "A noise-robust system for NIST 2012 speaker recognition evaluation," in *Interspeech-2013*, 2013, pp. 1981–1985.

[9] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *HLT '94 Proceedings of the workshop on Human Language Technology*, 1994, pp. 307–312.

[10] "NIST SRE12 evaluation plan," http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v11-r0.pdf.

[11] A. Stolcke, X. Anguera, K. Boakye, O. Cetin, A. Janin, M. Magimai-Doss, C. Wooters, and J. Zheng, "The SRI-ICSI spring 2007 meeting and lecture recognition system," in *Proc. NIST Rich Transcription Workshop, Springer Lecture Notes in Computer Science*, 2008, pp. 450–463.

[12] L. Deng, J. Li, J. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at Microsoft," in *ICASSP-2013*. IEEE, 2013, pp. 8604 – 8608.