Deep Neural Networks for extracting Baum-Welch statistics for Speaker Recognition

P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet and J. Alam

Centre de Recherche Informatique de Montréal (CRIM), Canada

{Patrick.Kenny, Vishwa.Gupta, Themos.Stafylakis, Pierre.Ouellet, Jahangir.Alam}@crim.ca

Abstract

We examine the use of Deep Neural Networks (DNN) in extracting Baum-Welch statistics for i-vector-based text-independent speaker recognition. Instead of training the universal background model using the standard EM algorithm, the components are predefined and correspond to the set of triphone states, the posterior occupancy probabilities of which are modeled by a DNN. Those assignments are then combined with the standard 60-dim MFCC features to calculate first order Baum-Welch statistics in order to train the i-vector extractor and extract i-vectors. The DNN-based assignment force the i-vectors to capture the idiosyncratic way in which each speaker pronounces each particular triphone state, which can enrich the standard short-term spectral representation of the standard i-vectors.

After experimenting with Switchboard data and a baseline PLDA classifier, our results showed that although the proposed i-vectors yield inferior performance compared to the standard ones, they are capable of attaining 16% relative improvement when fused with them, meaning that they carry useful complementary information about the speaker's identity. A further experiment with a different DNN configuration attained comparable performance with the baseline i-vectors on NIST 2012 (condition C2, female).

1. Introduction

Text-independent speaker recognition has been dominated by models that segment the input space on the basis of low-level acoustic events. The use of short-term spectral information (MFCC, PLP, etc.) that is augmented by Δ and $\Delta\Delta$ features, followed by a universal background model (UBM) that is trained without any phonetic information, and an i-vector for modelling whole utterances, has been proven to be the most robust front-end in distinguishing speakers in text-dependent speaker recognition. Yet, the use of the higher-level phonetic events as complementary to the acoustic ones has been advocated by several researchers to be beneficial, [1], [2].

Recently, Deep Neural Networks (DNN, [3], [4]) have clearly shown their superiority over Gaussian mixture models (GMM) for automatic speech recognition (ASR), with relative improvement in word error rate (WER) being about 30%, [5]. A fundamental difference between DNNs and GMMs (as well as earlier Neural Net approaches) is the capacity of DNNs in handling longer segments of speech as inputs (about 300ms), which enables them to make use of the information carried in the neighbourhood of the target-frame, in order to assign it probabilistically to one of the phonetic classes (usually triphone states).

Investigating ways to combine recent advances in deep architectures and speaker recognition can be a very promising direction of research. In [6], Deep Belief Networks (DBNs) are deployed in order to built an alternative i-vector extractor, that performs a non-linear transformation on the input features which produces the probability that an output unit is on, given the input features. In [7], a Boltzmann Machine was deployed as a back-end classifier and its performance was equivalent to a state-of-the-art PLDA model. In the case of DNNs for ASR, it has been recently demonstrated that conventional i-vectors can be very effective in fast speaker adaptation, when augmenting the input layer of the DNN with a speaker- or utterance-level i-vector, yielding a further $\sim$8% relative improvement in WER, [8].

In this paper, we show how a deep neural network can take the place of a universal background model (UBM) in collecting Baum-Welch statistics for text-independent speaker recognition with a conventional i-vector/PLDA architecture. Strictly speaking, talking about Baum-Welch statistics here is an abuse of language (we are not assuming that the data is generated by a Gaussian mixture model) but the term is appropriate since the statistics we extract are identical in form to traditional zero and first order Baum-Welch statistics. The only difference is in the way the posterior occupation probabilities are calculated for each frame.

Baum-Welch statistics extracted from a given frame give a sparse over complete representation in the sense of [9] . (The representation is over complete because representing a frame by a supervector of first-order statistics is redundant; it is sparse because, for a given frame, almost all of the occupation probabilities are zero.) Sparsity at the frame level makes for a very good representation at the utterance level for the purpose of text-independent speaker recognition. Acoustic events (say nasals and fricatives) occurring at different times in the course of the utterance will activate different components of the first-order statistics supervector. Thus they will not interfere with each other when the Baum-Welch statistics for individual frames are pooled together. This pooling yields a fixed dimensional representation of the utterance (the dimension is independent of the utterance duration) which greatly facilitates the task of developing back-ends. (Of course pooling loses information concerning the order of acoustic events but this is no drawback for text-independent speaker recognition.)

Our contribution in this paper then is to propose a slightly different sparse over complete representation for frames: one which captures information about acoustic-phonetic events (namely the pronunciation of individual triphones), rather than merely acoustic events (as in the case of a traditional speaker recognition system).

The rest of the paper is organized as follows. In Section 2, the main elements of DNNs for ASR are described. In Sec-

tion 3, we demonstrate how the DNN outputs are used in order to train the i-vector extractor and extract i-vectors. In Section 4, the experimental set-up is presented in depth and results on a subset of Switchboard are reported, followed by Section 5, where we present results on NIST SRE-2012.

2. Deep Neural Networks for speech recognition

The use of DNNs to model the emission probabilities in speech recognition has been revived during the last 5 years and several benchmark tests demonstrate their superiority over (both generatively- and discriminatively-trained) GMM-HMM systems, [4], [5].

2.1. Input and output layers of DNNs

A major difference between them and earlier NN implementations is the use of large window as input layer. For our principal experiment, we follow the DNN training recipe described in [5]. Assuming a frame of 20-30ms (augmented by its first Δ and second order $\Delta\Delta$ derivatives), a DNN uses as input layer not only the particular frame, but its neighbourhood (typically 5 left and 5 right), which forces the model to learn speech dynamics of longer time-spans. The frames are themselves LDA projections of 7 consecutive 13-dim static MFCCs (3 on each side), which can be considered as a data-driven analogue to the augmentation of the static MFCCs with Δ and $\Delta\Delta$. Cepstral mean subtraction is applied prior to LDA projection, where the means are estimated per conversation side. After the LDA projection, the features undergo a single semi-tied covariance transform, [10]. Finally, feature-space maximum likelihood linear regression (fMLLR) is applied, again estimated conversation side (Speaker Adaptive Training, SAT). For estimating the fMMLR transforms at runtime, a first pass of the data with a conventional HMM-GMM and language model is required for estimating the path in the lattice.

As a discriminative classifier, the DNN is trained in a such way that it provides posterior probability estimates about the HMM states $s \in \mathcal{S}$, given the observations, where $\mathcal{S}$ the set of triphone states. Assuming an observation o_{ut} that corresponds to the tth frame of an utterance u, the output of the DNN $y_{ut}(s)$ is given by the following expression

$$y_{ut}(s) = \frac{\exp \alpha_{ut}(s)}{\sum_{s'} \exp \alpha_{ut}(s')} \tag{1}$$

known as *softmax* activation function, where $\alpha_{ut}(s)$ the activation of the output layer corresponding to state s.

The remaining hidden layers are sigmoid, while it has been found that the optimal results for ASR are obtained with a 7-layer DNN, [5].

2.2. Training the DNN using cross-entropy

In order to train the DNN, the backpropagation algorithm is applied, with cross-entropy being a popular optimization criterion. The cross-entropy criterion minimizes the following objective function

$$\mathcal{J}_{CE} = -\sum_{u=1}^{U}\sum_{t=1}^{T_u} \log y_{ut}(s_{ut}) \tag{2}$$

where s_{ut} denotes the (known during training) state at time t of the uth utterance. The expression in eq. (2) is the cross-entropy between the multinomial distribution of the reference labels and the predictive distribution $y(s)$, which is continuous and defined on the simplex. It is worth noticing that by minimizing the entropy we also maximize the mutual information between outputs $y(s)$ and labels s, computed at the frame-level. Initial implementations of DNN training deployed stacked restricted Boltzmann Machines (RBMs) as an initialization step, that are pretrained in a greedy layer wise fashion using contrastive divergence. Such an initialization allows for the use of unlabelled data (all but the uppermost RBM do not require labels) and may prevent the model to stack in local maxima. More recent work, though (see [4]), has demonstrated that same results can be attained using the standard random initialization when the training data is large enough. Yet, in order to reproduce the results in [5] we initialize the DNN with the stacked-RBM method.

Table 1: ASR results (WER, %) on SWB and CHE

System	SWB	CHE	Total
GMM-HMM [5]	18.66	33.0	25.8
DNN [5]	14.2	25.7	20.0
DNN (ours)	15.6	27.4	21.5

2.3. Results on ASR using DNN

Our implementation, which follows the recipe of the baseline system described in [5] (with cross-entropy optimization criterion and frame-discriminative training), has been tested on Switchboard (SWB) and CallHome English (CHE) benchmarks. The results given in Table 1 show a marked improvement in Word Error Rate (WER) by using DNN instead of GMM-HMM. The GMM-HMM system is trained discriminatively, using Boosted Maximum Mutual Information (BMMI) as optimization criterion, which is considered as one of the most effective GMM-HMM training algorithms. Our DNN, compared to the implementation of DNN in [5] performs slightly worse, which can be attributed to the use of 5-hidden layer rather than 6.

We should finally note that in our implementation for speaker recognition, the HMM with language model is only used in the first pass in order to estimate the fMLLR transform. The DNN posteriors $y_{ut}(s)$ are estimated using (1) without the use of HMM or language model. Thus, the implied prior of the posterior is derived by the triphone state frequencies that appear on the training set. Although this implementation is not optimal for ASR, we considering it as a good starting point for the preliminary experiments on speaker recognition we present here.

3. Training and Extraction of i-vectors

The concept of i-vector extraction is based on the Factor Analysis extended to handle session and speaker variabilities of supervectors to Joint Factor Analysis (JFA), [11], [12], [13]. Contrary to JFA, different sessions of the same speaker are considered to be produced by different speakers. Rather than making distinction between the speaker and channel effects the total variability space in the i-vector extraction method simultaneously captures the speaker and channel variabilities, [14]. Given a C component GMM-UBM model λ with $\lambda_c = [w_c, m_c, \Sigma_c]$, $c = 1, 2, ..., C$ and an utterance having a sequence of T feature frames $y_1, y_2, ..., y_T$ the zero and first order Baum-Welch statistics on the (ancillary) UBM are computed as:

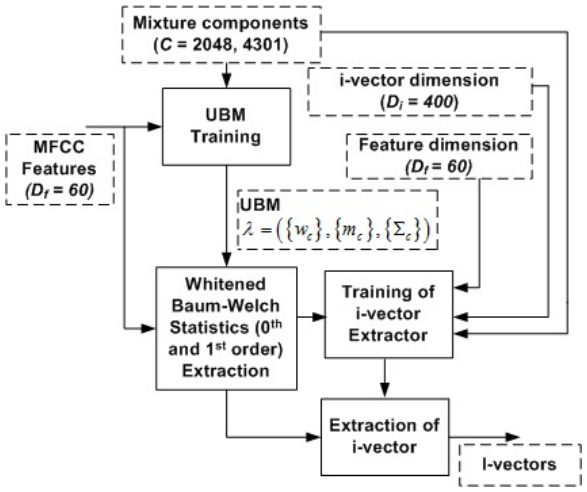


Figure 1: Block diagram showing different stages of I-vector extraction process

$$N_c = \sum_{t=1}^{T} \pi_t(c) \tag{3}$$

$$F_c = \sum_{t=1}^{T} \pi_t(c) y_t \tag{4}$$

where $\pi_t(c)$ the probabilistic alignment (posterior probability of the cth component for the tth observation) and $\pi_t(c) = P(c|y_t, \lambda)$ in the case of standard i-vectors.
For the proposed method, an ancillary UBM is needed, with parameters denoted again by λ. Its use in minor though, as it serves only to prewhiten the Baum-Welch statistics. Note that the Expectation-Maximization algorithm is not required for estimating λ. We simply use the probabilistic alignment $\pi_t(c)$ provided by the DNN on the training data to estimate it.

The Baum-Welch statistics are extracted using the following formula

$$F_c \leftarrow L_c^{-1}(F_c - N_c m_c) \tag{5}$$

where $L_c L_c^t = \Sigma_c$ the Cholesky decomposition of Σ_c.

The generative model for the i-vector can be expressed as:

$$M = m_c + T\theta,\ \theta \sim \mathcal{N}(0, I) \tag{6}$$

where M is a supervector constructed by appending together the first order statistics for each mixture component c, the columns of the low rank total variability matrix T span the subspace where most of the speaker specific information lives (along with channel effects). For each speech recording r, an i-vector i_r is obtained as the posterior expectation of θ:

$$i_r = \hat{\theta} = (I + T^t \Sigma^{-1} N(r) T)^{-1} T^t \Sigma^{-1} F(r) \tag{7}$$

where $N(r)$ is a diagonal matrix of dimension $CD_f \times CD_f$ whose diagonal blocks are $N_c I, (c = 1, 2, ..., C)$, D_f is the feature dimension, $F(r)$ is a supervector of dimension $CD_f \times 1$ obtained by concatenating all first centred order Baum-Welch statistics F_c, diagonal covariance matrix Σ is of dimension $CD_f \times CD_f$ and it models the residual variability not captured by the total variability matrix T.

4. Experiments on Switchboard

4.1. Training Data

In this section, we present our primary experiment on Switchboard. We chose Switchboard because of its transcribed data in Switchboard 1 that is needed in order to train the DNN. We used Switchboard 1 Release 2 (SW1R2) and Switchboard 2, Phases I, II and III (SW2PH1, SW2PH2, and SW2PH3). Training the UBM was done with SW1R2 only; training the i-vector and PLDA models was done using SW1R2, SW2PH1 and SW2PH2. SW2PH3 was set aside as a development/test set.

4.2. Voice Activity Detection (VAD)

We used a GMM-based VAD with two GMMs, similar to that described in [15]. The 44-dimensional VAD MFCC features were used to train the two (background and speech) 256-component diagonal covariance GMMs. The features were calculated from telephone-only audio data from the NIST 2004 to 2010 SREs. We produced one raw segmentation per conversation side using this VAD.

4.2.1. Echo Cancellation

Switchboard corpora, especially SW1R2, contain significant amounts of cross-talk. In order to remove such cross-talk from a conversation side, we followed a recipe provided by the Brno University of Technology, to produce what we call post-processed segmentations:

1. Calculate the per-frame log-energy for both sides of the conversation under consideration.
2. Normalize the log-energy, over the whole length of the conversation and for both sides separately.
3. Using the raw segmentation for the conversation side of interest, calculate, for each segment, the average normalized log-energy (ANLE) for both sides of the conversation.
4. For each segment, if the ANLE of that segment is lower than a certain threshold, label that segment as silence; if the ANLE of the conversation side is 3 dB or more below the ANLE of the opposite (interlocutor's) conversation side, label that segment as silence; otherwise, preserve the raw segment label.

Figure 1 presents a block diagram showing various steps of the i-vector extraction process from the MFCC (mel-frequency cepstral coefficients) features. An i-vector extractors of dimension $D_i = 400$ was trained using features computed from the Switchboard 1 Release 2 (SW1R2) and Switchboard 2, Phases I, II (SW2PH1, SW2PH2) database. Finally, the ancillary UBM was trained on the SW1R2 database.

4.3. Evaluation set

For evaluation, we have created a test set with single enrolment utterances and all possible gender-dependent pair combinations from Switchboard 3, yielding 12354 and 4441264 target and non target trials, respectively, for female speakers, and 9867 and 2636867 target and non target trials for male.

4.4. Baseline System

4.4.1. Front-End Features

The front-end features were 19 MFCC calculated over a 25 ms window every 10 ms, plus a log-energy coefficient, giving a 20-dimensional vector; Δ and $\Delta\Delta$ coefficients were then added to form one 60-dimensional vector per 10 ms, each of which was then Gaussianized over a 3-second window.

4.4.2. UBM

The baseline gender-independent UBM was trained using all of the speech from SW1R2, as determined by the post-processed VAD segmentations. First, a single-Gaussian diagonal GMM (the sample mean vector and diagonal covariance matrix) was generated, which was then iteratively doubled in size and re-estimated until the diagonal-covariance GMM comprised 2048 components. A full-covariance UBM was then produced from the diagonal-covariance UBM, and was then re-estimated.

4.5. DNN posteriors

4.5.1. Front-End Features

The exact same front-end features were used as in the baseline system (see Section 4.4.1).

4.5.2. Ancillary UBM

The UBM differences between the DNN system and the baseline system are:

- For each speech data frame, a top-20 list of mixture component indices and their corresponding posterior probabilities were produced using the DNN.
- The ancillary UBM, with full covariance matrices, was directly trained in one pass using the posteriors from the DNN and the front-end features.
- The resulting ancillary UBM had 4301 components instead of 2048 for the baseline UBM.

4.5.3. PLDA model

As backend classifier, a generative PLDA model is utilized. The i-vectors are prewhitened and projected onto the unit-sphere (a transform known as length-normalization within the speaker recognition community, [16]) that is used in order to make the i-vector distribution more Gaussian-like and less heavy-tailed. PLDA assumes the following generative model

$$i_r = \mu + V y_s + \epsilon_r \tag{8}$$

where μ the global mean, V the rectangular matrix whose columns span the speaker variability space, $y_s \sim N(0, I)$ the vector of speaker factors (fixed for each speaker s) and $\epsilon_r \sim N(0, \Sigma_\epsilon)$ the residual, with covariance matrix Σ_ϵ. After experimentation, we found that for the standard i-vectors, the optimal dimensionality of y_s was $\dim(y_s) = 120$, while for the proposed i-vectors the full-rank speaker variability setting was superior, i.e. $\dim(y_s) = \dim(i_r)$.

4.6. Experimental Results

Apart from the baseline and the DNN based i-vectors, we are also reporting results after fusing the LLRs of the two systems, using the Bosaris toolkit, [17].

Table 2: Results on Switchboard - female speakers

method	EER (%)	minNDCF$_{08}$	minNDCF$_{10}$
Baseline	2.38	0.097	0.361
NeuralNet	3.47	0.140	0.495
Fusion	2.13	0.083	0.320

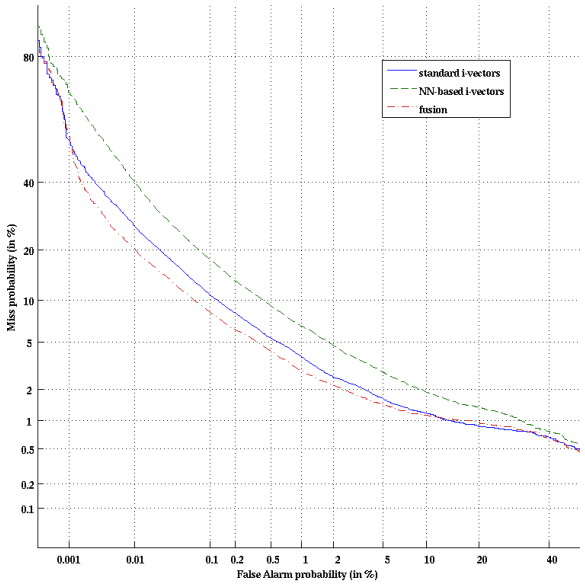
Table 3: Results on Switchboard - male speakers

method	EER (%)	minNDCF$_{08}$	minNDCF$_{10}$
Baseline	1.74	0.074	0.319
NeuralNet	2.31	0.081	0.357
Fusion	1.53	0.059	0.272

The results are given in Tables 2 and 3, in terms of Equal Error Rate (EER) and minimum normalized Detection Cost Function (minNDCF) of NIST '08 and '10. From these tables, it becomes evident that we did not manage to outperform the standard i-vectors. This is in line with most of the systems that attempt to model the phonetic events rather than the acoustic, [1], [2]. Yet, the DNN-based i-vectors seem to fuse well with the standard ones, reducing the (averaged among genders) minimum normalized DCFs of '08 and '10 by 17% and 13%, respectively. Figure 2 and 3 show the DET curves (derived with the Bosaris toolkit, [17]). We clearly observe that for all operating points of practical interest the fused LLRs outperformed the baseline on both genders.

We should also note that we tried to increase the dimensionality of DNN-based i-vectors from 400 to 800 but with negligible improvement. Finally, we attempted to fuse the two systems on the PLDA domain, by concatenating the two i-vectors, yet, the score-level fusion was significantly superior.

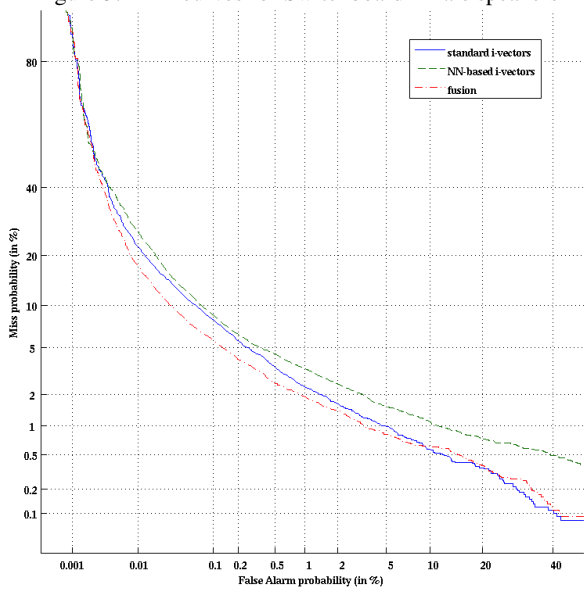
Figure 2: DET curves for Switchboard - female speakers



5. Experiments on NIST 2012

After this paper was submitted, we were informed that the same idea has recently been explored and accepted for publication, [18]. Moreover, the authors report an impressing 30% rela-

Figure 3: DET curves for Switchboard - male speakers



tive improvement on NIST-2012 telephone data, even without applying fusion with the standard i-vectors. Apart from some minor differences (they did not use $\Delta\Delta$ coefficients and the number of triphone states was 3500, instead of 4301), the major differences between their implementation and ours are summarized below.

- About 1300 hours of speech for DNN training (English telephone speech from Switchboard, Callhome and Fisher), compared to 278 hours that we used.
- The use of 40-dimensional filter-bank with 15 frames, instead of the standard MFCC feature with 10 frames followed by LDA.

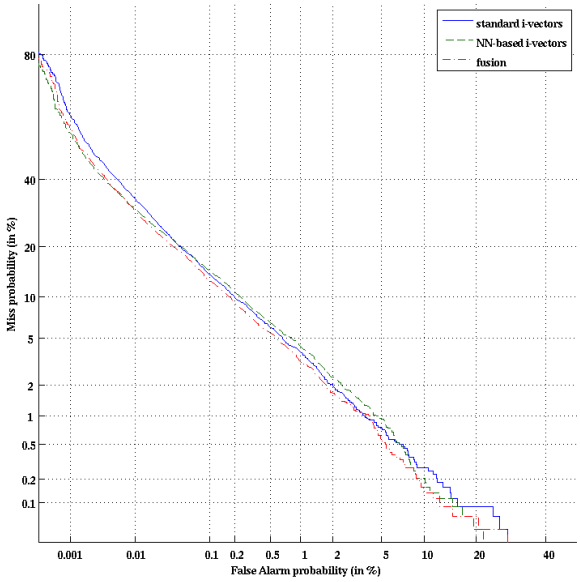
Given the fact that improvements of this range are rare in speaker recognition, we tried to replicate their results. The training set we used train our new DNN was composed of about 1200 hours and was identical to the one used in [18], excluding the Callhome dataset. We have used identical training and evaluation sets (namely 1040 speaker models for evaluation, 4432 target and 8356128 nontarget trials). The only major difference was on the number of triphone states. We used only 429 triphone states instead of 3500, due to computational/temporal constrains.

Our first step was to replicate the results attained in [18] using a baseline i-vector/PLDA model with a 2048-component UBM. After experimentation, we concluded that a full-rank PLDA (i.e. with $\dim(y_s) = 400$) was optimal. Comparing Table 4 (line 1) with the corresponding results in [18] we conclude that they are close enough, at least in terms of DCFs. For a fair comparison with our DNN system, a 512-component UBM was also used for the standard i-vector/PLDA baseline. The results on NIST 2012 (C2, female) are given in Table 4 and Fig. 4. We observe that the proposed method has a comparable performance with the baseline, and is better in the low-false alarm area. Finally, when fusing the the 512-component baseline with the DNN-based system a further improvement was attained.

Table 4: Results on NIST-2012 (C2 condition) - female speakers

method	EER (%)	minNDCF$_{08}$	minNDCF$_{10}$
Baseline-2048	1.58	0.086	0.381
Baseline-512	1.95	0.108	0.434
NeuralNet	2.16	0.112	0.400
Fusion	1.81	0.099	0.398

Figure 4: DET curves for NIST-2012 (C2 condition) female speakers



6. Conclusions and future work

We proposed the use of Deep Neural Networks (DNN) in extracting Baum-Welch statistics for i-vector-based text-independent speaker recognition. We do so in order to obtain a model that focuses on phonetic events, rather that the usual short-term acoustic ones. On top, an i-vector extractor was trained and used to extract i-vectors, followed by a generative PLDA model. The experiments on NIST demonstrated a 16% relative improvement after fusing their scores with the ones of PLDA model using standard i-vectors.

As future directions, we plan to experiment with several DNN configurations, such like sequence-discriminative training and different optimization criteria (e.g. [5]) and evaluate it on different datasets. The very successful implementation of the same approach in [18], together with the experiments that we performed on NIST-2012 show that it has the potential to become the new state-of-the-art model for text-independent speaker recognition.

7. References

[1] T. Kinnunen and H. Li, *An overview of text-independent speaker recognition: From features to supervectors*, Speech Communication, vol. 52, no. 1, pp. 12-40, 2010.

[2] M. Diez, L J Rodriguez-Fuentes, M. Penagarikano, A. Varona, G. Bordel, *Using Phone Log-Likelihood Ratios as Features for Speaker Recognition*, in Proceedings of Interspeech 2013, Lyon, France, 2013.

[3] 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*, IEEE Signal Processing Magazine, vol. 28, no. 6, pp. 82-97, 2012.

[4] L. Deng, G. Hinton, B. Kingsbury, *New types of Deep Neural Network Learning for speech recognition and related applications: An overview*, in Proceedings ICASSP 2013, pp. 8599-8603, 2013.

[5] K. Vesely, A. Ghoshal, L. Burget and D. Povey, *Sequence-discriminative training of deep neural networks*, in proceedings of Interspeech 2013.

[6] V. Vasilakakis, S. Cumani and P. Laface, *Speaker recognition by means of Deep Belief Networks*, Biometric Technologies in Forensic Science, Nijmegen, 14-15 October 2013.

[7] T. Stafylakis, P. Kenny, M. Senoussaoui and P. Dumouchel, *Preliminary investigation of Boltzmann machine classifiers for speaker recognition*, in Proceedings of Odyssey Speaker and Language Recognition Workshop, 2012.

[8] V. Gupta, P. Kenny, P. Ouellet and T, Stafylakis, *I-vector based speaker adaptation of Deep Neural Networks for french broadcast audio transcription*, (to appear) in proceedings of ICASSP 2014.

[9] Y.-W. Teh, M. Welling, S. Osindero and G. Hinton, *Energy-based models for sparse overcomplete representations*, The Journal of Machine Learning Research, vol. 4, pp. 1235-1260, 2003.

[10] M. J. F. Gales, *Semi-tied covariance matrices for hidden Markov models*, IEEE Trans. on Audio Speech and Language Processing, vol. 7, no. 3, pp. 272-281, May 1999.

[11] P. Kenny, G. Bouliane, P. Ouellet, and P. Dumouchel, *Joint Factor Analysis versus Eigenchannels in Speaker Recognition*, IEEE Trans. on Audio Speech and Language Processing, vol. 15, no. 4, pp. 1435-1447, May 2007.

[12] P. Kenny, G. Bouliane, P. Ouellet, and P. Dumouchel, *Speaker and session variability in GMM-based speaker verification*, IEEE Trans. on Audio Speech and Language Processing, vol. 15, no. 4, pp. 1448-1460, May 2007.

[13] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, *A Study of Inter-Speaker Variability in Speaker Verification*, IEEE Trans. on Audio Speech and Language Processing, vol. 16, no. 5, pp. 980-988, July 2008.

[14] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, *Front-End Factor Analysis For Speaker Verification*, IEEE Trans. on Audio Speech and Language Processing, vol. 19, no. 4, pp. 788-798, May 2011.

[15] L. Ferrer, Y. Lei, M. McLaren, N. Scheffer, Martin Graciarena, and Vikramjit Mitra, *SRI 2012 NIST Speaker Recognition Evaluation System Description*, 2012.

[16] Daniel Garcia-Romero and Carol Y. Espy-Wilson, *Analysis of i-vector Length Normalization in Speaker Recognition Systems*, in proceedings of Interspeech 2011.

[17] N. Brummer and E. de Villiers, *The BOSARIS Toolkit User Guide: Theory, Algorithms and Code for Binary Classifier Score Processing*, Tech. Rep., 2011. [Online]. Available: https://sites.google.com/site/nikobrummer

[18] Y. Lei, N. Scheffer, L. Ferrer and M. McLaren, *A novel scheme for speaker recognition using a phonetically-aware Deep Neural Network*, (to appear) in proceedings of ICASSP 2014.