Speaker adaptation of DNN-based ASR with i-vectors: Does it actually adapt models to speakers?

Mickael Rouvier, Benoit Favre

Aix Marseille Universite, CNRS, LIF UMR 7279

firstename.lastname@lif.univ-mrs.fr

10.21437/Interspeech.2014-503

Abstract

Deep neural networks (DNN) are currently very successful for acoustic modeling in ASR systems. One of the main challenges with DNNs is unsupervised speaker adaptation from an initial speaker clustering, because DNNs have a very large number of parameters. Recently, a method has been proposed to adapt DNNs to speakers by combining speaker-specific information (in the form of i-vectors computed at the speaker-cluster level) with fMLLR-transformed acoustic features. In this paper we try to gain insight on what kind of adaptation is performed on DNNs when stacking i-vectors with acoustic features and what information exactly is carried by i-vectors. We observe on REPERE corpus that DNNs trained on i-vector features concatenated with fMLLR-transformed acoustic features lead to a gain of 0.7 points. The experiments shows that using i-vector stacking in DNN acoustic models is not only performing speaker adaptation, but also adaptation to acoustic conditions.

1. Introduction

In Automatic Speech Recognition (ASR), the adaptation of acoustic models consists in creating new specialized acoustic models from general models given homogeneous data (for instance a speaker cluster). Adaptation tends to reduce the divergence between the initial acoustic model and a corpus generally closer to the test corpus. In the literature, several adaptation methods have been proposed, either using a different corpus (Maximum A Posteriori - MAP [1]), or from a subcorpus (Maximum Likelihood Linear Regression - MLLR [2]). In a multi-pass transcription, other methods allow unsupervised adaptation by using a first transcription.

Deep Neural Networks (DNN) have recently been successful for acoustic modeling and have lead to significant improvements over earlier state-of-the-art approaches such as Gaussian Mixture Model (GMM) Hidden Markov Models (HMM) on several ASR tasks [3, 4, 5]. Nevertheless, one of the main challenge with DNNs is unsupervied speaker adaptation within homogeneous speaker segments. Indeed, portability approaches, such as MLLR or MAP which work very well for GMMs, can not be applied to DNNs. Unlike GMMs, DNNs do not have a clear and identifiable structure, and DNNs have significantly more parameters due to large and deep hidden layers. Therefore, adaptation of so many parameters to so few data is not straightforward. Different methods have been proposed in the literature for speaker adaptation. The methods fall into two categories : parameters adaptation of a DNN or stacking of additional speaker-dependent parameters in the input of the DNN.

In [6], the authors propose to adapt the bias of top layers (the outputs of the final hidden layer) with an affine transformation. Another method in [7] proposes to adapt the DNN weights conservatively by forcing the distribution estimated from the adapted model to be close to that estimated from the weights before adaptation. This constraint is implemented by adding Kullback-Leibler Divergence (KLD) regularization to the adaptation criterion. In [8] the authors propose to stack i-vector features with fMLLR features as the input in the DNN. I-vectors are supposed to capture all the speaker-specific information in a reduced number of features. They are commonly used in speaker verification and speaker identification and yield state of the art performance in these tasks. The main idea is to allow the DNN to learn from the speaker-specific information. In a multi-pass system, the authors propose to extract i-vectors from fMLLR-normalized acoustic features (estimated in the first-pass of the ASR system).

Unfortunately, fMLLR is a technique supposed to remove speaker variability. So the i-vectors extracted from the fMLLR-normalized acoustic features should no longer contain information related to the speaker. In our work, we try to gain insight on what kind of adaptation is performed on DNNs when stacking i-vectors with acoustic features and what information exactly is carried by i-vectors. In a first experiment, we propose to use different acoustic parameters focused on different areas of the acoustic space. The i-vectors, used in conjunction with acoustic features should therefore model different kind of information. In a second experiment we try to get insight on the information modeled by i-vectors by partitioning the i-vectors and checking whether the partitions match natural classes of the data. The third experiment is designed towards understanding the impact of speaker diarization on adaptation. For this experiment, we extract i-vectors according to the different clusterings (-only or full diarization) and see how this impacts word error rate. Finally, we remove speaker-related information from the i-vectors (by using surrogate i-vectors) to see if the DNN is adapting to speakers or not.

The papers is organized as follows: Section 2 summarizes the i-vector approach. Section 3 presents the methods used for integrated i-vectors in DNN-based ASR. The results of our experiments are explained in Section 4. Section 5 concludes with a discussion of possible directions for future works.

2. I-Vectors

The i-vector is a low-dimensional feature that characterizes speakers. Our goal is to extract an i-vector for each speaker and stack this feature with acoustic features in the input of a DNN performing acoustic modeling in an ASR system.

2.1. Extraction

I-vector approaches have become the state-of-the-art in the speaker verification field. They provide an elegant way of reducing a large-dimensional input data to a small-dimensional feature vector, while at the same time retaining most of the rel-

evant information. The technique was originally inspired by the Joint Factor Analysis (JFA) framework introduced in [9].

Given a GMM, the corresponding mean super-vector M can be approximated by:

$$M = m + Tw \quad (1)$$

where m is the mean super-vector taken from a GMM-UBM trained on a large number of speakers; T is a low-rank rectangular matrix spanning the subspace covering the relevant variability; w is a low-dimensional vector with a normally distributed prior $N(0, \mathbf{I})$. After iteratively estimating matrix T over a training corpus, equation 1 allows to use the lower-dimensional vector w as a speaker model in place of a large GMM. w is referred to as an i-vector. The i-vector algorithm is fully described in [10].

2.2. Normalization

I-vectors are extracted on records that are treated as being statistically independent (regardless of the association between all variability sources). The i-vectors are projected on a total variability space, no distinction is made between the sources of variability.

At this step the i-vectors contain both useless and useful information. Some normalization techniques have been proposed in order to remove useless information [11, 12]. We propose to use the Eigen Factor Radial (EFR) algorithm that is a generalization of length normalization. This method is an iterative process with two goals:

1. Ensure that the i-vectors are distributed among $\mathbf{N}(0, \mathbf{I})$. One consequence of that constraint is that the vector dimensions of i-vectors are mutually independent.
2. Apply length normalization to the i-vectors to make the test and trial i-vector distributions more similar and more Gaussian shaped.

In the training corpus, for each speaker we extract an i-vector. The goal of the normalization algorithm is to compute parameters for the i-vectors present in the training corpus and apply these parameters to the i-vectors present in the test corpus.

Algorithm 1 describes the training method for the i-vector normalizing parameters. The parameters (the mean μ_i and the covariance matrix Σ_i) of the i-vectors present in the training corpus are saved at each iteration i (*step 0*). Next, the i-vectors are conditioned using the parameters of the current iteration: *step 1* is the classical data standardization, and *step 2* is length normalization.

Algorithm 1: Normalization algorithm of i-vectors on the training corpus

for $i = 1$ *to* $nb\_of\_iterations$ **do**
  Step 0: Compute the mean μ_i and the covariance matrix Σ_i on the training corpus;
  for *each* w *in the training corpus:* **do**
    Step 1: $w = \Sigma_i^{-\frac{1}{2}} (w - \mu_i)$;
    Step 2: $w = \frac{w}{\|w\|}$;
  end
end

On the test corpus, after BIC (Bayesian Information Criterion) clustering, an i-vector is computed for each cluster. The i-vectors are then normalized iteratively, in a manner similar to that used during the training phase, as explained in algorithm 2. The difference lies in the absence of *step 0*: the mean μ_i and covariance matrix Σ_i used for each iteration in this phase are the ones saved during the training phase. As in the training phase, *step 1* is the data standardization, and *step 2* is length normalization.

Algorithm 2: Normalization algorithm for the test phase

for $i = 1$ *to* $nb\_of\_iterations$ **do**
  Step 1: $w = \Sigma_i^{-\frac{1}{2}} (w - \mu_i)$;
  Step 2: $w = \frac{w}{\|w\|}$;
end

3. Integrating i-vectors with DNN

In ASR, a DNN is used as acoustic model instead of GMMs. DNN is a neural network with several hidden layers. In our experiments, the DNN has 4 hidden layers. The output layer is a soft-max layer, and the outputs represent the log-posterior of the output labels, which are context-dependent HMM states (there are about 7,000 states in our experiments). The number of neurons in the hidden layer is the same for all hidden layers: 1536 neurons. The nonlinearities in the hidden layers are tanh functions. The objective function is the cross-entropy criterion, i.e. for each frame, the log-probability of the correct class. The weights are updated using mini-batches of size 256 frames.

The acoustic features used are derived by processing the conventional 13-dimensional PLP. The features are stacked across ±4 frames to produce 117 dimensional vectors. A Linear Discriminant Analysis (LDA) is used to reduce the dimensionality to 40. Context-dependent HMM states are used as classes for the LDA estimation. A fMLLR is applied to normalize inter-speaker variability. Then the features are stacked across ±4 frames to produce 360 dimensional vectors. The procedure for integrating i-vector with DNNs is as follows. First, we extract for each speaker cluster an i-vector (the notion of speaker cluster is defined in Section 4.2). Then, the i-vector is concatenated to every acoustic frame of the cluster in order to form the input of the DNN.

4. Experiments and results

4.1. Corpus

The data used for the experiments are those of the REPERE 2013 evaluation campaign [13]. The data is composed of a subset from 68 TV shows recorded from French TV channels BFM and LCP. The corpus contains broadcast news videos, political discussions and street interviews. Only a part of the recordings are annotated, giving a total duration of 10 hours.

4.2. Speaker Diarization

Speaker diarization is carried out using the LIUM open-source speaker diarization toolkit [14]. First a speaker segmentation is performed to detect fine-grained speaker changes using Generalized Likelihood Ratio (GLR). Then a hierarchical agglomerative clustering is used to group the segments belonging to the same speakers using the BIC distance.

4.3. ASR

In our experiments we used the Kaldi ASR toolkit [15]. The speech transcription process is carried in two passes (additional gains can be obtained by performing more passes, but our experiments are restricted to those passes for the sake of clarity):

1. The first pass: A first automatic transcription is performed with a GMM-HMM model. The model is composed of 7,000 states and 150,000 Gaussians.
2. The second pass: The word-graphs output by the first pass are used to compute a fMLLR transform on each cluster given by the speaker diarization. Then, the second pass is performed using a DNN trained on acoustic feature on which we apply fMLLR transformation [2].

The acoustic models are trained using a set of data from distinct sources. The training corpus is composed of 227 hours of wide-band recordings (167h from ESTER 1 and 2 campaign and 60h from EPAC [16, 17]). The language-model is based on trigram models and is composed of 95k words. Different textual data from multiple sources are used to train language-model : the audio corpus transcript, the french gigaword [18] and additional data collected from the Web. To estimate and interpolate these models, the SRILM [19] toolkit is employed using modified Knser-Ney discounting without cut-off.

4.4. i-vector

Matrix T of equation 1 is estimated over the training corpus. The matrix is iteratively estimated using the Expectation Maximization (EM) algorithm. We have chosen a dimension of 100 for the i-vectors. The GMM-UBM is composed of 512 Gaussians computed using the ALIZE speaker recognition toolkit[1]. We propose to extract different kinds of i-vectors, that differ with respect to acoustic features they model:

- i-vector $\Delta\Delta$: the acoustic feature is a 13-dimension PLP, augmented by first and second derivatives. This configuration is the one that is closest to what is used in speaker verification.
- i-vector LDA: the 13-dimension PLP are staked across ± 4 frames to produce 117 dimensional vectors. Then LDA is applied to reduce the dimensionality to 40. The context-dependent HMM states are used as classes for the LDA estimation.
- i-vector LDA+fMLLR: is based on LDA features as described previously with an additional fMLLR transform applied to normalize inter-speaker variability. The fMLLR is estimated using the first pass of our speech transcription system.

The different features proposed focus on different zones of the acoustic space. The i-vectors will therefore model different kind of information. The *i-vector* $\Delta\Delta$ system focuses on speaker-information, whereas *i-vector LDA* and *i-vector LDA+fMLLR* systems should focus more on additional information.

4.5. Speaker adaptation or not?

In this section, we describe experiments exploring the speaker-adaptation effects of stacking i-vectors with acoustic features as input of the DNN.

Acoustic features

[1] http://alize.univ-avignon.fr/

Table 1 shows the results obtained on the REPERE corpus using the different kinds of i-vectors. All i-vectors were normalized with three iterations of the EFR algorithm. The baseline consists in not stacking i-vectors for speaker adaptation.

	Sub	Del	Ins	WER
Baseline	11.63	6.61	2.77	21.01
i-vector $\Delta\Delta$	11.26	6.33	2.89	20.48
i-vector LDA	11.17	6.27	2.88	20.32
i-vector LDA+fMLLR	11.19	6.24	2.89	20.31

Table 1: Results obtained on the REPERE corpus using different kinds of i-vectors.

The *baseline* system (corresponding to pass-2) obtains 21.01% of Word Error Rate (WER). The best configuration is the *i-vector LDA+fMLLR* system that corresponds to extracting i-vectors from acoustic features normalized by LDA and fMLLR. We observe that the i-vectors which follow the speaker verification reciepe (*i-vectors* $\Delta\Delta$) do not obtain the best results. The System using i-vectors focused on more than just speaker information obtains the best results therefore the DNN is adapting to something else than just speaker information.

Bi-partition criteria

We want to check if i-vectors extracted on different acoustic features model different (or similar) information. We propose to bi-partition i-vectors extracted from the training corpus (ie 5908 i-vectors) and see which data criteria (acoustic classes) best partition them compared to an automatic bi-partition. The automatic bi-partition is performed using the K-Means algorithm (with $K = 2$) and the cosine distance. For each criterion we compute a correct classification rate according to the criterion on the automatic bi-partition given by K-means.

We define three criteria: (1) Gender: male or female; (2) Music: musical background or not (3) Noise: presence of various noises annotated in the transcript, or not. We remind the reader that an i-vector is extracted on every speaker-cluster. Table 2 shows the correct classification rate obtained according different criteria (*Gender, Music and Noise*). We observe that the *Gender* criterion obtains the best correct classification rate on the i-vectors extracted from standard acoustic features $\Delta\Delta$ (77.27%). The correct classification rate decreases when the acoustic features are normalized by LDA (65.46%) and LDA+fMLLR (54.39%). These results confirms that LDA and fMLLR methods reduce speaker variability which is not captured anymore by i-vectors.

	Gender	Music	Noise
i-vector $\Delta\Delta$	77.27	47.15	46.70
i-vector LDA	65.46	49.42	45.70
i-vector LDA+fMLLR	54.39	52.50	51.83

Table 2: Correct classification rate according to acoustic classes.

We observe the opposite effect on *Music* and *Noise*. The correct classification rate increases by stacking LDA and fMLLR transforms. These results can be explained by the fact that *Music* and *Noise* (and more generally the acoustic condition) tend to be the most discriminant criteria. The acoustic condition is a difficult variability to compensate for in DNNs. I-vectors are extracted from the cluster and seem to yield more acoustic context to be used by the DNN.

Speaker-clustering

Traditionally, in speaker diarization for ASR systems a first clustering is performed with the BIC distance while complete diarization systems typically include additional grouping steps. The BIC clustering allows to group segments belonging to similar acoustic conditions, generally containing the speech of a single speaker. In order to merge the multiple clusters of the same speaker, Diarization systems perform an additional pass with a Normalized Cross Likelihood Ratio (NCLR) based on bottom-up clustering. The NCLR bottom-up clustering is performed on the clusters obtained after BIC segmentation.

We propose to analyze three differents systems: (1) *i-vector Segment* : there is no clustering (this system obtained a Diarization Error Rate (DER) of 84.41%), (2) *i-vector BIC* : the segments are grouped according to acoustic closeness (this system obtained a DER of 27.21%) and (3) *i-vector BIC+NCLR* : the segments are grouped according to speakers (this system obtained a DER of 16.14%). For all theses systems we use i-vectors extracted from acoustic features normalized by LDA and fMLLR. Table 3 summarizes the WER of the ASR system according to the clusters used for computing i-vectors. We observe that the system *i-vector BIC* obtains the best results 20.31% WER whereas the system *i-vector BIC+NCLR* obtains 20.40% WER. Theses results suggest that clusters modeling the acoustic condition provide most information to the DNN.

	Sub	Del	Ins	WER
i-vector Segment	12.23	7.12	3.78	23.13
i-vector BIC	11.19	6.24	2.89	20.31
i-vector BIC+NCLR	11.11	6.65	2.63	20.40

Table 3: Results obtained on the REPERE corpus using different clustering methods.

Remove speaker information

In this experiment, our goal is to completely remove speaker-related information from the i-vectors in order to understand whether the DNN is being adapted or not to the speaker. We propose for each i-vector present in the test corpus to replace it with the nearest i-vector present in the training corpus which does not belong to that speaker (effectively, the i-vector is carying a different identity). The distance used is a cosine distance between i-vectors. Then, we propose to decode the test set using i-vectors replaced by those i-vectors from the training corpus (*i-vector Train*). In table 4, we observe that the *i-vector Train* system has a gain of 0.32 compare to the *Baseline* system. Although the cosine distance finds a similar speaker, most of the speaker identity is removed. We believe that the gains obtained are mostly due to the similar acoustic conditions.

	Sub	Del	Ins	WER
Baseline	11.63	6.61	2.77	21.01
i-vector Train	11.45	6.29	2.95	20.69
i-vector Test	11.19	6.24	2.89	20.31

Table 4: Results obtained on the REPERE corpus using i-vectors transplanted from a different speaker.

Discussion

We observed in the bi-parition criteria experiment, that i-vectors extracted on acoustic features normalized by LDA and fMLLR contained less speaker variability than i-vectors extracted from standard parametrization. We also observed that the information related to acoustics conditions is more important on normalized acoustic features than on a standard parametrisation. In the speaker-clustering experiment, using different clustering algorithms, we observed that the best results are obtained using i-vectors extracted according to the acoustic condition (*i-vector BIC*) rather than i-vectors extracted on speaker clusters (*i-vector BIC+NCLR*). In addition, we observed, in the experiment where we remove speaker information, that a gain can be obtained by using i-vectors of the training corpus *i-vector Train* compared to *Baseline*. All of these experiments show that using i-vector stacking with DNNs is not only performing speaker adaptation, but also adaptation to acoustic conditions. Understanding how to model speaker and acoustic conditions seems to be critical in improving ASR performance, and more work is necessary in order to leverage all the techniques developed in the speaker verification community. Indeed, relevant information is different in ASR, where the objective is to obtain speaker-independent models, than it is for speaker identification which focuses on speaker-specific traits. For example, some techniques have been proposed in speaker recognition to tackle irrelevant information in i-vectors [20, 21, 22]. It is now well established that the limitations of the i-vector representation of speech segments have started to become apparent. The sensitivity of i-vectors to segment durations is an obvious case and different approaches have been proposed to take it into account [23]. Therefore, it seems very interesting to study the approaches proposed by the speaker recognition community in order to adapt them in ASR.

4.6. Impact of i-vector normalization

Table 5 shows the results obtained with i-vectors with different levels of normalization. All the results are reported using i-vectors extracted on acoustic features normalized by a LDA and a fMLLR. We observe that disabling i-vector normalization does not significantly improve the results relative to the *baseline* system (20.96% WER - *i-vector no-norm.*). A first iteration of the algorithm allows to improve the results by about 0.6 points (*i-vector EFR iter1*). And the others iterations do not improve nor deteriorate the results. In speaker verification, in [20], the best results are obtained in using two iterations. We think that a first iteration is necessary for the DNN in order to decorrelate the feature space. But the other iterations are not necessary because the DNN can decorrelate by itself.

	Sub	Del	Ins	WER
i-vector no-norm.	11.55	6.61	2.81	20.96
i-vector EFR iter1	11.16	6.31	2.81	20.30
i-vector EFR iter2	11.15	6.27	2.88	20.30
i-vector EFR iter3	11.19	6.24	2.89	20.31

Table 5: Results obtained on the REPERE corpus using or not EFR normalization.

5. Conclusions

In this paper we study speaker adaptation in DNNs by stacking i-vectors with acoustic features as input to the model. I-vectors extracted from acoustic feature are complementary and provide additional gains when used in conjunction with acoustic features as input to the DNN. In the experiments we observe that the DNN is not only performing speaker adaptation, but also adaptation to acoustic conditions.

6. References

[1] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 291–298, 1994.

[2] M. J. Gales and P. Woodland, "Mean and variance adaptation within the mllr framework," *Computer Speech & Language*, vol. 10, no. 4, pp. 249–264, 1996.

[3] A.-R. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5060–5063.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[5] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *Interspeech*, 2011, pp. 437–440.

[6] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 366–369.

[7] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7893–7897.

[8] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 55–59.

[9] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1435–1447, 2007.

[10] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification." in *INTERSPEECH*, vol. 9, 2009, pp. 1559–1562.

[11] P.-M. Bousquet, D. Matrouf, and J.-F. Bonastre, "Intersession compensation and scoring methods in the i-vectors space for speaker recognition." in *InterSpeech*, 2011, pp. 485–488.

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

[13] O. Galibert and J. Kahn, "The first official repere evaluation," in *Speech, Language and Audio for Multimedia (SLAM 2013)*, 2013.

[14] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization." in *InterSpeech*, 2013.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[16] S. Galliano, G. Gravier, and L. Chaubard, "The ester 2 evaluation campaign for the rich transcription of french radio broadcasts." in *Interspeech*, vol. 9, 2009, pp. 2583–2586.

[17] Y. Esteve, T. Bazillon, J.-Y. Antoine, F. Béchet, and J. Farinas, "The epac corpus: Manual and automatic annotations of conversational speech in french broadcast news." in *LREC*, 2010.

[18] A. Mendonça, D. Graff, and D. DiPersio, "French gigaword," 2009.

[19] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit." in *InterSpeech*, 2002.

[20] P.-M. Bousquet, A. Larcher, D. Matrouf, J.-F. Bonastre, and O. Plchot, "Variance-spectra based normalization for i-vector standard and probabilistic linear discriminant analysis," in *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2012.

[21] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Speaker and Language Recognition Workshop (IEEE)*, 2010.

[22] M. Senoussaoui, P. Kenny, N. Brümmer, E. De Villiers, and P. Dumouchel, "Mixture of plda models in i-vector space for gender-independent speaker recognition." in *InterSpeech*, 2011, pp. 25–28.

[23] P. Kenny, T. Stafylakis, P. Ouellet, and M. J. Alam, "Jfa-based front ends for speaker recognition."