JFA for Speaker Recognition with Random Digit Strings

Themos Stafylakis[1], *Patrick Kenny*[1], *Jahangir Alam*[1] *and Marcel Kockmann*[2]

[1]Centre de Recherche Informatique de Montreal (CRIM), Quebec, Canada
[2]VoiceTrust, Ontario, Canada
themos.stafylakis@crim.ca

Abstract

In this paper, we examine the use of Joint Factor Analysis methods on RSR2015 digits. A tied-mixture model is used for segmentation of the utterances into digits, while Joint Factor Analysis and a Joint Density model are deployed for features and backend, respectively. A novel approach for digit-dependent fusion of UBM-component log-likelihood ratios is introduced, yielding the best results so far. The fusion of 5 different JFA features gives an equal-error rate of 3.6%, compared to 6.3% attained by the a baseline GMM-UBM model with score normalization.

Index Terms: Joint Factor Analysis, text-dependent, speaker recognition

10.21437/Interspeech.2015-82

1. Introduction

Over the last few years, an increasing interest in text-dependent speaker recognition is observed in literature and industry. Encouraged by the tremendous increase in performance attained in the text-independent setting, the potential use of text-dependent speaker recognition as a natural and flexible means for biometric authentication is being reexamined. A key-ingredient that makes text-dependent speaker recognition such an interesting biometric field is the very short utterances that are required to attain low error rates, as a result of the constrained phonetic content that the user is prompted to utter during enrollment and test, [1], [2].

One of the variants of text-dependent speaker recognition is the case where the user is prompted to utter a digit strings, randomly chosen by the system. This variant differs from the typical text-dependent speaker recognition setting where the phrase is fixed for each user and hence, enrollment and test phrases are matched. In the random digit strings scenario, test utterances contain a subset of the digits that appear in the enrollment phase, where usually the speaker is prompted to utter all of 10 digits several times. The fact that the level of granularity becomes the digit makes the application of a segmentation algorithm and the use of segment-level vectors a necessity for enabling the back-end model to make decisions based on pairs of segments of matched phonetic content.

For the RSR-digits task, the authors in [9] proposed a HMM system called HiLAM model. It is a standard HMM likelihood based approach, where each speaker is modelled by a HMM and each state correspond one of the 10 digit. The standard Viterbi algorithm is deployed for evaluating the likelihood of the model given the new test utterance and the digit string it contains. Moreover, rather that directly estimating the emission probabilities using ML estimates, the authors proposed a hierarchical training approach, where the enrollment utterances are initially used to adapt the UBM and create an initial speaker model, followed by a second adaptation phase where state emission probabilities are iteratively estimated, using Viterbi training and MAP adaptation with the speaker model as a prior.

The approach we present here shares a few common elements with the HiLAM model, mainly the use of the HMM-Viterbi machinery to segment the utterances. The most substantial difference is the use of Joint Factor Analysis (JFA) methods which enables us to apply e.g. channel compensation and subspace modelling, [12]. Moreover, instead of evaluating JFA by-the-book, we are rather using it as a feature extractor ([6], [8], [7]) but in a different way than the familiar i-vectors, [13]. The features are subsequently passed to a trainable backend, which we call *Joint Density Backend* (JDB), where the joint density of two features belonging to the same speaker is estimated using target trials from the training set, and the corresponding one of the non-target is derived by appying the independence assumption. The JFA-JDB approach can be considered as an analogue to the state-of-the-art for text-independent speaker recognition i-vector/PLDA paradigm, well suited to the text-dependent setting and more flexible for subspace modelling.

The rest of the paper is organized as follows. In Sect. 2, we show how to segment the data using a tied-mixture model. The JFA features are discussed one by one, with details on how the corresponding JFA models are trained. In Sect. 3, the JDB is explained, while a novel method for applying digit-dependent fusion of component LLRs is also introduced. Finally, in Sect. 4, the experimental results on RSR-digits are presented and compared to a baseline system.

2. Segmentation, JFA training and feature extraction

2.1. UBM, TMM and segmentation

We are starting by training a gender independent UBM trained on Mixer data with $N_c = 128$ components. We are then using the RSR-digit background set (*bkg*) to adapt the NIST UBM using iterative MAP adaptation (means only). Note that both the initial and the adapted UBMs will be used to extract JFA features. We start with a NIST UBM in order to train one of our JFA models on NIST.

To segment the RSR data into digits, the codebook of the adapted UBM (i.e. means and covariance matrices) is kept fixed and the weights are used to distinguish between the 10 digits. This model is known as Tied Mixture Model (TMM). To train the weights vectors, the Viterbi training algorithm is applied, where each utterance is initialized with uniform segmentation. It is an EM algorithm, where the optimal path for all training utterances is estimated on the E-step using hard assignments (via Viterbi algorithm), and the model parameters (i.e. the weights for each of the 10 digits) are reestimated given the paths, on the

M-step. Note that the digit strings for all utterances are given and used by the algorithms during training and recognition, as defined by the RSR2015 protocol.

2.2. JFA features

As discussed in our previous works on text-dependent speaker recognition, we have decided to use JFA due to its increased flexibility, [6], [8], [7]. It can be used to extract features which are channel compensated and can either be *local* or *global*, low or supervector dimensional. Local is a feature that models a segment (and in this paper a digit) while global a feature that models the whole utterance. Low dimensional features are termed $\boldsymbol{y}$-vectors while the *supervector dimensional* $\boldsymbol{z}$-vectors, so that they correspond to the usual JFA naming convention. The general equation of JFA is as follows

$$\boldsymbol{s}^r = \boldsymbol{m} + D\boldsymbol{z} + V\boldsymbol{y} + U\boldsymbol{x}^r \tag{1}$$

where $\boldsymbol{s}^r$ is the (unobserved) supervector of the rth utterance of a speaker, $\boldsymbol{m}$ the concatenated means of the UBM, $\boldsymbol{z}$ and $\boldsymbol{y}$ are variables shared across all utterances of the same speaker and $\boldsymbol{x}^r$ a variable that models channel effects. Finally, V and U are low-rank matrices while D a diagonal. Assume now that the utterances are segmented into digits, we end-up with several possible combinations of features, which are discuss in detail below.

2.2.1. Local $\boldsymbol{y}$-vectors with RSR-JFA

The first $\boldsymbol{yx}$-JFA model with local $\boldsymbol{y}$-vectors is trained on the *bkg* set of RSR digits (i.e. 97 speakers). It is therefore perfectly matched to the *dev* and *eval* set, both in terms of phonetic content and channel effects. Note that for JFA training using local vectors, speaker variables are tied across all segments in the training set of the same speaker and digit. On the other hand, channel factors are always unique for each recording and shared across segments of the same recording. Eq. (2) shows how each supervector that corresponds to a recording r and digit d is assumed to be generated.

$$\boldsymbol{s}^{r,d} = \boldsymbol{m}_{rsr} + V\boldsymbol{y}^d + U\boldsymbol{x}^r \tag{2}$$

Finally, recall that the TMM is composed of a common codebook and different set of weights per digit. Therefore, for each set of segments, the corresponding digit-dependent weight vector is the one used for training the JFA and extracting $\boldsymbol{y}$-vectors. This applies to all local vectors that we presented below.

2.2.2. Local $\boldsymbol{y}$-vectors with NIST-JFA

A second $\boldsymbol{yx}$-JFA model is trained, this time on NIST (Mixer data). We emphasize that no segmentation of NIST data is performed on NIST data, and therefore the JFA model is trained in the usual way, as eq. (3) shows.

$$\boldsymbol{s}^r = \boldsymbol{m}_{nist} + V\boldsymbol{y} + U\boldsymbol{x}^r \tag{3}$$

It is interesting to note the high mismatch in duration between training and test $\boldsymbol{y}$-vectors utterances (2min NIST recording and 150-300ms digit). It is compulsory though to use utterances of long duration in order to model robustly correlations between UBM components, [3]. The use of short segments in JFA training of the local $\boldsymbol{y}$-vectors with RSR digits discussed above is probably the only exemption to this rule, because of the matched phonetic content (and channel effects) that the training set exhibits with the runtime utterances. On the other hand, in cases where no short utterances of matched phonetic content are available for training subspace models, training with long utterances seems to be the only way.

Moreover, note that while the UBM used to train JFA is the NIST UBM, the extraction of $\boldsymbol{y}$-vectors from RSR is performed using the TMM. What enables us to do so it the fact that the TMM is derived from the NIST UBM, using mean-only MAP adaptation, i.e. an operation which preserves the correspondence between components of the two UBMs. Thus, to extract local $\boldsymbol{y}$-vectors with the NIST-JFA, we use again eq. (2) where in this case $\{V, U\}$ refer to the NIST-JFA parameters. Note that since the feature extraction is performed with the TMM, the mean in the JFA equation should be $\boldsymbol{m}_{rsr}$.

2.2.3. Global $\boldsymbol{y}$-vectors

We have also experimented with global $\boldsymbol{y}$-features. The results were very poor, though, and therefore we do not report them in this paper. The most probable reason for their failure is the unmatched phonetic content between enrollment and test utterances (all 10 digits vs. a random set of 5 digits). Uncertainty modelling might be helpful in such a case, but we did not attempt it in this paper, [5]. Eq. (4) shows the corresponding generative model.

$$\boldsymbol{s}^r = \boldsymbol{m}_{rsr} + V\boldsymbol{y} + U\boldsymbol{x}^r \tag{4}$$

2.2.4. Local $\boldsymbol{z}$-vectors

Local $\boldsymbol{z}$-features are trained on *bkg* set. We trained a $\boldsymbol{zx}$-JFA model with channel factors tied across segments of the same utterance, as always. For $\boldsymbol{z}$-vectors, the relevance factor was empirically set equal to 2 without any attempt to optimize it. The generative model is given in Eq. (5).

$$\boldsymbol{s}^{r,d} = \boldsymbol{m}_{rsr} + D\boldsymbol{z}^d + U\boldsymbol{x}^r \tag{5}$$

2.2.5. Global $\boldsymbol{z}$-vectors

Finally, $\boldsymbol{z}$-features are again trained on *bkg* set. As in the case of global $\boldsymbol{y}$-vectors, the segmentation of the utterances into digits is of no use, since both speaker and channel factors are tied across segments of the same utterance. The relevance factor was set equal to 2, as in the case of local $\boldsymbol{z}$-vectors and the generative model is as follows

$$\boldsymbol{s}^r = \boldsymbol{m}_{rsr} + D\boldsymbol{z} + U\boldsymbol{x}^r \tag{6}$$

3. Backend model and component fusion

3.1. Joint Density Model for backend

The Joint Density Backend (JDB) was originally proposed in [11] in the context of text-dependent speaker recognition with averaging of the enrollment utterances. Two are the main changes we introduce here. The first is to make the model asymmetric, to allow different distributions between enrollment and test vectors. The second is to extend it to $\boldsymbol{z}$-vectors, where the likelihood function is factorized across UBM components.

The JDB is trained on the *bkg* set of RSR-digits. To do so, a set of enrollment models needs to be created, using 3 10-digit utterances from the same speaker and handset. We do that in order to replicate the protocol defined for the *dev* and *eval* set. Then, a list of target trials (same speaker, different channel) is created, by pairing the models with the 5-digit utterances from the *bkg* set.

In the case of $\boldsymbol{y}$-vectors, training the backend is straightforward. We are simply calculating first and second order statistics of the training trial list, after concatenating enrollment $\boldsymbol{y}_e$ and test $\boldsymbol{y}_t$ feature vectors into a single vector $\boldsymbol{\phi}$ of double dimensionality, i.e. $\boldsymbol{\phi}^T = [\boldsymbol{y}_e^T, \boldsymbol{y}_t^T]$. When local vector are used, each segment of the test utterance is paired with the corresponding one in the enrollment side, based on the digit. Thus, each trial contributes as many $\boldsymbol{\phi}$ vectors as the number of digits in the test utterances (i.e. 5). The model parameters of the JDB are estimated as follows.

$$\boldsymbol{\mu} = E[\phi],\ C = E[\boldsymbol{\phi}\boldsymbol{\phi}^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T = \left[\begin{array}{cc} C_{ee} & C_{et} \\ C_{te} & C_{tt} \end{array} \right] \quad (7)$$

where $C_{te} = C_{et}^T$.

The likelihood ratio of the JDB consists of evaluating the pdf of two gaussians of the same mean $\boldsymbol{\mu}$ and different covariance matrices. For the numerator we have $C_{num} = C$, while for the denominator C_{den}, defined as follows

$$C_{den} = \left[\begin{array}{cc} C_{ee} & 0 \\ 0 & C_{tt} \end{array} \right] \quad (8)$$

i.e. we set the crosscorrelation terms C_{et} equal to zero, using the independence assumption under different speaker hypothesis. We mention that after experimentation with full and diagonal versions of the submatrices C_{ee}, C_{tt} and C_{et}, full matrices performed better, apart from the case of global $\boldsymbol{z}$-vectors, where diagonal sub matrices exhibited superior performance. This is due to the fact that for global $\boldsymbol{z}$-vectors, 5 times less trials are available for training the JDB, compared to the local $\boldsymbol{z}$-vectors, i.e. the number of digits in the test utterances. When local vectors are used, the 5 LLRs are added to form the LLR of the trial, as the model suggests.

In the case of $\boldsymbol{z}$-vectors, component specific JDB parameters are estimated $\{\boldsymbol{\mu}_c, C_c\}_{c=1}^{N_c}$ during training, using only those supervector entries that correspond to each component c. During evaluation, the component specific LLRs are estimated and summed, due to independence assumption. For the most complicated case of local $\boldsymbol{z}$-vectors, the overall LLR l of a trial is a summation of LLRs l^d for each digit d in the test utterance, while each l^d is itself a summation of LLRs l_c^d, one for each mixture component c. Hence, the overall LLR l of a trial becomes as double summation, as follows

$$l = \sum_{d \in t} l^d = \sum_{d \in t} \sum_c l_c^d \quad (9)$$

where $d \in t$ means those 5 digits contained in test utterance t.

3.2. Fusion of component LLRs

The method we describe here applies only to the case of local $\boldsymbol{z}$-vectors. The idea is to weight the LLRs l_c^d in a digit-dependent way, using the following linear fusion model.

$$l^d = \sum_c w_c^d l_c^d + b^d \quad (10)$$

As before, the LLR of the overall trial will be the summation of the LLRs of each digit. To estimate the fusion weights $\boldsymbol{w}^d = \{w_c^d\}_{c=1}^{N_c}$ and biases b^d, the *dev* set was used. After experimentation we found that L_2 regularization was very helpful. This is due to the high dimensionality of $\boldsymbol{w}^d$. Note that $w_c^d \in \Re$ and therefore some of the weights may be negative. L_2 regularization is applied simply by placing $0 \leq \lambda \leq 1$ in the following update formula of the weights $\boldsymbol{w}$

$$\boldsymbol{w} \leftarrow (1 - \lambda)\boldsymbol{w} - \alpha \Delta_{\boldsymbol{w}} f(\boldsymbol{w}) \quad (11)$$

where $f(\boldsymbol{w})$ the objective function without regularization to be minimized and α the learning rate. We set $\lambda = 0.03$ after some not exhaustive experimentation. Note that $\lambda = 0$ corresponds to no regularization.

Set	Gender	# target	# nontarget
Dev	M	5134	251381
Dev	F	4886	224714
Eval	M	5359	300969
Eval	F	5188	248852

Table 1: Trial statistics for RSR Digits per set and gender

3.3. Mean and Length Normalization

For local vectors, the digit dependent means are estimated from the *bkg* set and subtracted from all vectors. A different mean is estimated for enrollment end test utterances. Then, length normalization is applied, simply by projecting the vectors onto the unit-sphere, without prewhitening. In the case of local vectors the projection is performed for each segment separately.

3.4. Score Normalization

Despite the use of a probabilistic backend, score normalization is required in order to reduce the error rates especially in the case of $\boldsymbol{z}$-vectors. There is no need to extract new enrollment and test utterances though. The enrollment vectors for TNorm and test vectors for ZNorm have already been extracted for training the JDB, as explained above. Note that score normalization is gender dependent, and in fact the only gender dependent operation used in this paper. Score normalization is applied as usual, i.e. on LLRs of the whole trial. We mention though the possibility in the case of local vectors of first applying score normalization on digit LLRs, and adding them afterwards. The T- and Z-normalized LLRs are finally added together to form S-normalized LLRs.

4. Experimental Results

In the RSR protocol on Digits (RSR2015, Part 3), each speaker model contains 3 10-digit utterances, recorded with the same handset. The digits are uttered in a random order, as defined by the prompt. Each test utterance contains a random set 5 digits. For all types of utterances, the set of digits is known to the system and the algorithm can use it, as we do in the case of local vectors. The number of trials are given in Table 1 for each set and gender.

The experiments were conducted using 60-dimensional PLP with mean and variance normalization, extracted using HTK. After applying Voice Activity Detection to each of the utterances, those of less than 1s remaining duration or of SNR below 15 dB were rejected. Moreover, speaker models with less than 3 enrollment utterances were excluded from the lists.

For baseline, we are reporting results using GMM-UBM with SNorm (Table 2). This system uses exactly the same configuration (front-end, VAD, UBM) with the proposed one. Note that the notation within matrix entries means male/female.

Set	EER (%)	DCF_{08}	DCF_{10}
Dev	4.81/8.04	0.217/0.356	0.595/0.775
Eval	4.15/8.36	0.224/0.383	0.798/0.874

Table 2: Results using GMM-UBM. The notation means male/female

feat	G/L	EER (%)	DCF_{08}	DCF_{10}
$\boldsymbol{y}$-nist	L	6.56/7.33	0.289/0.377	0.737/0.825
$\boldsymbol{y}$-rsr	L	5.70/6.23	0.244/0.326	0.659/0.799
z	G	4.85/7.60	0.219/0.353	0.605/0.775
z	L	5.50/6.73	0.245/0.344	0.694/**0.756**
z	L_f	**4.40/5.60**	**0.201/0.309**	**0.631**/0.766
fusion		3.49/4.30	0.163/0.244	0.550/0.688

Table 3: Results on RSR Digits, *dev* set. The notation means male/female and L_f refers to fusion of component LLRs.

The results are significantly inferior to those attained in the RSR2005 - Part 1, where fixed phrases are used instead of random digit strings, [9], [15]. Moreover, it is known that likelihood-based systems (e.g. GMM-UBM, HiLAM) perform well in RSR2015 (at least on Part 1), mainly due to the exceptionally low channel effects that the database contains, [9]. Nevertheless, the proposed JFA-based framework has demonstrated its capacity in dealing with datasets of much richer channel effect, where model-based channel compensation is compulsory, [7].

The results on *dev* and *eval* sets with the proposed method are given in Table 3 and 4, respectively. We did not include the results using global $\boldsymbol{y}$-vectors, since the EER was above 20%. In general, $\boldsymbol{z}$-vectors perform better compared to $\boldsymbol{y}$-vectors, underlying the difficulties of speaker subspace methods in text-dependent speaker recognition, [4], [10], [14], [5]. Moreover, global $\boldsymbol{z}$-vectors seem to be equivalent to the local ones, and it is only after component LLR fusion is applied that the latter yield clearly better performance (denoted by L_f). Recall though that local vectors use half the information contained in the enrollment side, i.e. 5 out of 10 digits. As a final step, we are fusing all the 5 JFA systems, using the *dev* set for estimating fusion weights. The fused system performs very well, as the EER averaged across genders drops to around 3.6% on the eval set, which is a very good result compared to the benchmark. Both minDCF metrics exhibit a significant improvement, too, which underlies the complementary information conveyed by the several JFA features. Finally, the DET curves are plotted for each of the two genders in the *eval* set in Fig. 1 and Fig. 2. The improvement obtained by fusing all JFA systems is clearly depicted.

feat	G/L	EER (%)	DCF_{08}	DCF_{10}
$\boldsymbol{y}$-nist	L	4.57/8.06	0.224/0.371	0.706/0.822
$\boldsymbol{y}$-rsr	L	4.18/7.38	0.204/0.341	0.641/0.787
z	G	4.08/6.97	0.200/0.336	0.640/0.774
z	L	3.87/6.90	0.191/0.319	0.618/0.749
z	L_f	**3.25/6.08**	**0.167/0.291**	**0.565/0.744**
fusion		2.65/4.54	0.136/0.229	0.530/0.660

Table 4: Results on RSR Digits, *eval* set, the notation means male/female and L_f refers to fusion of component LLRs.

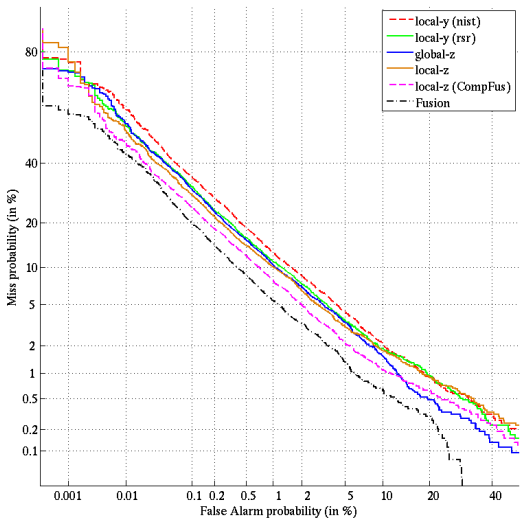

Figure 1: Results on RSR Digits, Male - *eval* set

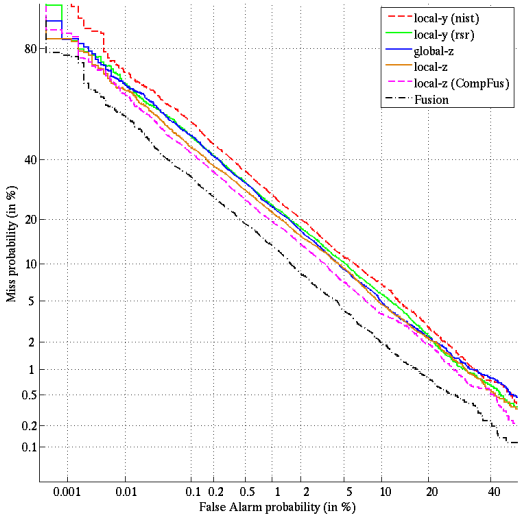

Figure 2: Results on RSR Digits, Female - *eval* set.

5. Conclusions

In this paper, we examined the applicability of our recently proposed JFA-JDB approach to RSR2015 digits. The HMM formulation, quite common to text-dependent speaker recognition, fits even more naturally in the case of random digit string, if we desire to compare segments of the same digit. We did so by deploying a Tied Mixture Model for the emission probabilities, so that each digit is modelled by its own set of weights and the codebook is shared across digits. Given the segmentation obtained by Viterbi algorithm, we used JFA to extract features that are either global and local, of small or of supervector size, all of which are channel compensated. A trainable backend model is performing the final LLRs calculations, while fusion of the LLRs of each feature is also applied. Moreover, in the case of local $\boldsymbol{z}$-vectors, a novel fusion method that applies to LLRs of each UBM component was introduced, yielding the best results. After fusing all 5 different systems, an average EER of 3.6% is attained, that corresponds to 40% relative improvement with respect to the baseline GMM-UBM with SNorm.

6. References

[1] M. Hébert, "Text-dependent speaker recognition", in *Springer Handbook of Speech Processing*, pages 743-762. Springer-Verlag, Heidelberg, 2008.

[2] Tomi Kinnunen and Haizhou Li "An overview of text-independent speaker recognition: From features to supervectors", in *Speech Communication*, pages 12-40, Volume 52 Issue 1, January, 2010.

[3] H. Aronowitz and O. Barkan, "On leveraging conversational data for building a text dependent speaker verification system," *Interspeech* 2013.

[4] H. Aronowitz and A. Rendel, "Domain Adaptation for Text-Dependent Speaker Recognition," *Interspeech* 2014.

[5] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, and M. Kockmann, "Text-dependent speaker recognition using PLDA with uncertainty propagation," *Interspeech* 2013.

[6] P. Kenny, T. Stafylakis, P. Ouellet, and M. J. Alam, "JFA-based front ends for speaker recognition," *ICASSP* 2014.

[7] P. Kenny, T. Stafylakis, M. J. Alam, and M. Kockmann,"JFA modelling with left-to-right structure and a new back-end for text-dependent speaker recognition," *ICASSP* 2015.

[8] P. Kenny, T. Stafylakis, M. J. Alam, P. Ouellet and M. Kockmann, "Joint Factor Analysis for Text-Dependent Speaker Verification," *Odyssey* 2014.

[9] A. Larcher, A.-K. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015", *Speech Communication*, March 2013.

[10] A. Larcher, K.-A. Lee, B. Ma, and H. Li, "Phonetically constrained PLDA modeling for text-dependent speaker verification with multiple short utterances," *ICASSP* 2013.

[11] Sandro Cumani, and Pietro Laface, "Generative pairwise models for speaker recognition" , *Odyssey* 2014.

[12] P. Kenny "Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms," Tech. Rep., 2005 [Online]. Available: http://www.crim.ca/perso/patrick.kenny

[13] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Trans. ASLP*, 2011.

[14] S. Novoselov, T. Pekhovsky, A. Shulipa, A. Sholokhov, "Text-dependent GMM-JFA system for password based speaker verification", *ICASSP* 2014.

[15] A. Miguel, J. Villalba, A. Ortega, E. Lleida and C. Vaquero, "Factor Analysis with Sampling Methods for Text Dependent Speaker Recognition," *Interspeech* 2014.