Factor Analysis for Speaker Segmentation and Improved Speaker Diarization

Brecht Desplanques, Kris Demuynck, Jean-Pierre Martens

ELIS Multimedia Lab
Ghent University - iMinds, Belgium
brecht.desplanques@ugent.be

10.21437/Interspeech.2015-106

Abstract

Speaker diarization includes two steps: speaker segmentation and speaker clustering. Speaker segmentation searches for speaker boundaries, whereas speaker clustering aims at grouping speech segments of the same speaker. In this work, the segmentation is improved by replacing the Bayesian Information Criterion (BIC) with a new iVector-based approach. Unlike BIC-based methods which trigger on any acoustic dissimilarities, the proposed method suppresses phonetic variations and accentuates speaker differences. More specifically our method generates boundaries based on the distance between two speaker factor vectors that are extracted on a frame-by-frame basis. The extraction relies on an eigenvoice matrix so that large differences between speaker factor vectors indicate a different speaker. A Mahalanobis-based distance measure, in which the covariance matrix compensates for the remaining and detrimental phonetic variability, is shown to generate accurate boundaries. The detected segments are clustered by a state-of-the-art iVector Probabilistic Linear Discriminant Analysis system. Experiments on the COST278 multilingual broadcast news database show relative reductions of 50% in boundary detection errors. The speaker error rate is reduced by 8% relative.

Index Terms: speaker change detection, speaker diarization, clustering, segmentation, factor analysis

1. Introduction

Speaker diarization systems deal with the "who-spoke-when?" problem. The objective is to assign a speaker label to every speech segment (sentence). Speaker diarization encompasses both speaker segmentation and speaker clustering. The segmentation stage splits the audio stream into homogenous segments, whereas the clustering stage groups the generated segments into clusters. Each cluster corresponds with a particular speaker. In this paper we focus on improving the segmentation stage because inaccurate and inserted segment boundaries can have a detrimental effect on clustering: short segments do not provide enough data to estimate reliable speaker models while non-homogeneous segments make clustering harder. Furthermore, prior speaker information may be available e.g. in the form of television show scripts and hence more advanced segmentation techniques that can exploit this extra information are called for.

During the initial speaker change detection we replace the popular Bayesian Information Criterion (BIC) [1] by our proposed algorithm which triggers on real speaker differences by suppressing phonetic variability. The new method, which is described in Section 4, extracts a fixed number of speaker factors for each frame using a sliding window approach. The speaker factors are extracted using a speaker variability matrix, comparable to the iVector paradigm [2]. At speaker boundaries we expect the speaker factors to change. A Mahalanobis-based distance measure is used to detect these changes. The distance measure is designed to compensate for undesirable speaker factor changes caused by varying phonetic content.

The actual clustering of the segments is performed by the two-step Agglomerative Hierarchical Clustering (AHC) system proposed in [3]. In this approach an initial BIC clustering stage is followed by iVector Probabilistic Linear Discriminant Analysis (PLDA) clustering [3]. The proposed systems are evaluated on the COST278 multilingual broadcast news data set [4]. We evaluate the boundary accuracy before and after clustering. We also study the general diarization performance by looking at the speaker error rate.

2. BIC-based speaker segmentation

In this work we focus on the speaker segmentation performance and we therefore start from oracle speech/non-speech marks. Non-speech segments longer than 1s are discarded and all continuous speech segments are analyzed separately. Every 10ms we extract 16 MFCCs and a normalized log-energy [5]. A two-stage speaker segmentation algorithm is used to detect the homogeneous speaker turns as proposed in [6].

2.1. Boundary generation

Candidate change points are generated at places of maximum difference between the statistical distribution of the acoustic vectors in two windows (N_w frames) to the left and right of the candidate boundary position. The distance measure is defined as the log-likelihood ratio:

$$D_{\text{LLR}}(t) = 2\log|\mathbf{\Sigma}_{L+R}| - \log|\mathbf{\Sigma}_L| - \log|\mathbf{\Sigma}_R| \quad (1)$$

with each $\mathbf{\Sigma}$ the Maximum Likelihood (ML) full covariance of the acoustic features in the left (L), right (R) and merged (L+R) window.

To avoid detection of spurious peaks we average the LLR values across a window of N_{avg} frames. For each speech segment $\mathcal{S}$ the $N_p(\mathcal{S})$ largest peaks are selected, with N_p proportional to the number of frames $N_f(\mathcal{S})$ in $\mathcal{S}$:

$$N_p(\mathcal{S}) = \max(N_{p,min}, \left\lceil \frac{N_f(\mathcal{S})}{r} \right\rceil) \quad (2)$$

$N_{p,min}$ is the minimum number of peaks to detect and r is the presumed minimum duration of the speaker turns when all turns would be of equal length. We also enforce a real and shorter minimum duration of 1s for each speaker turn during the selection process.

2.2. Boundary elimination

The boundary generation stage produces many false positives which cannot be eliminated by our simple peak detection algorithm without losing too many real boundaries. The initial set of boundary positions is pruned by agglomerative clustering of adjacent speaker turns based on their acoustic similarity given by ΔBIC:

$$\Delta BIC = (N_L + N_R)\log|\mathbf{\Sigma}_{L+R}| - N_L \log|\mathbf{\Sigma}_L| - N_R \log|\mathbf{\Sigma}_R| - \lambda P \quad (3)$$

where N and $\mathbf{\Sigma}$ are the number of frames and full covariance matrix of the corresponding windows respectively. Note that the windows have a variable length at this stage. P is a penalty term

$$P = \frac{1}{2}\left(d + \frac{1}{2}d(d+1)\right)\log(N_L + N_R) \quad (4)$$

with d the dimension of the feature vectors. Inside each continuous speech segment we merge the most similar adjacent speaker turns with the lowest ΔBIC value and update the ΔBIC values that are affected by this merge. This process is iterated until the stopping criterion is met ($\min \Delta BIC > 0$). Parameter λ in (3) controls the number of eliminated boundaries.

3. Agglomerative clustering

The detected speaker turns are merged using the two-stage Agglomerative Hierarchical Clustering (AHC) approach from [3].

3.1. Initial BIC clustering

In the first stage, the clusters still contain little data, and hence robust techniques are needed. We therefore use BIC-based clustering. Whereas the boundary elimination only looked at the acoustic similarity via ΔBIC between adjacent pairs of speaker segments, we now compare all pairs.

3.2. iVector extraction

In the second stage the clusters contain enough data to apply more advanced techniques. First, unwanted variation such as noise and channel is suppressed. A Frame selection module [5] retains the high-energetic frames only. These frames should be the least affected by background noise. In addition the features of the selected frames of each cluster are normalized by means of Feature Warping [7].

Next, iVector PLDA is used to iteratively merge clusters. The main idea is to analyze the different sources of variability between clusters (speaker, channel, phonetic content,...) as the speaker clustering should obviously focus on the variability that can be owed to speaker changes. We use Total Variability (TV) [2] modeling to initialize the variability analysis. This approach tries to model as much variability as possible in a low dimensional subspace. A low rank matrix T, called the TV matrix or the iVector extractor, is used to approximate the GMM mean supervector $\boldsymbol{m}_c$ of cluster c as

$$\boldsymbol{m}_c = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{x}_c \quad (5)$$

where $\boldsymbol{m}$ is the supervector of the Universal Background Model (UBM) of speech. $\boldsymbol{x_c}$ is the fixed length iVector that contains all relevant information concerning cluster c. The procedure for extracting iVectors is described in [8]. The prior distribution of the iVectors is assumed to be a standard normal distribution. The TV matrix $\boldsymbol{T}$ is learned form a large data corpus by means of Principal Component Analysis (PCA) initialization [9] followed by a number of iterations of the non-simplified Expectation-Maximization algorithm described in [8].

3.3. PLDA clustering

Now we consider another factor analysis model to extract the speaker-specific information from the iVectors. As the iVectors $\boldsymbol{x}_c$ are of sufficiently low dimension we can achieve this via the modified PLDA framework [10]. After whitening and length normalization [11] each iVector is modeled as

$$\boldsymbol{x}_c = \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{y}_c + \boldsymbol{\epsilon}_r \quad (6)$$

where $\boldsymbol{\mu}$ is a global offset and $\boldsymbol{V}$ provides the basis for the speaker-specific subspace. $\boldsymbol{y}_c$ is a MAP point estimate of the latent variable $\boldsymbol{y}$ which has a standard normal distribution. The residual term $\boldsymbol{\epsilon}_r$ models the nuisance variability and it is assumed to be Gaussian with zero mean and full covariance $\mathbf{\Sigma}$.

The scores during AHC clustering can now be computed as the log-likelihood ratio for a hypothesis test

$$\mathrm{LLR}_{\mathrm{PLDA}}(c_i, c_j) = \log \frac{p(\boldsymbol{x}_{c_i}, \boldsymbol{x}_{c_j}|\mathcal{H}_s)}{p(\boldsymbol{x}_{c_i}|\mathcal{H}_d)p(\boldsymbol{x}_{c_j}|\mathcal{H}_d)} \quad (7)$$

where $\mathcal{H}_s$ is the hypothesis that clusters c_i and c_j are uttered by the same speaker, $\mathcal{H}_d$ assumes different speakers. We can remove the global offset $\boldsymbol{\mu}$ from all iVectors as it will have no impact on the LLR score. The LLR can now be evaluated as

$$\mathrm{LLR}_{\mathrm{PLDA}}(c_i, c_j) = \boldsymbol{x}_{c_i}^T \boldsymbol{Q}\boldsymbol{x}_{c_i} + \boldsymbol{x}_{c_j}^T \boldsymbol{Q}\boldsymbol{x}_{c_j} + 2\boldsymbol{x}_{c_i}^T \boldsymbol{P}\boldsymbol{x}_{c_j} \quad (8)$$

where matrices $\boldsymbol{Q}$ and $\boldsymbol{P}$ solely depend on the total variability $\mathbf{\Sigma}_{tot} = \boldsymbol{V}^T\boldsymbol{V} + \mathbf{\Sigma}$ and the inter speaker variability $\mathbf{\Sigma}_{inter} = \boldsymbol{V}^T\boldsymbol{V}$. For more details see [11].

The procedure for extracting iVectors relies on zero- and first-order statistics generated by the UBM [8]. When the two most similar clusters are being merged we generate a common iVector by summing up these sufficient statistics and re-extracting the new $\boldsymbol{x}_c$. The clustering process is terminated when the scores stop exceeding a predetermined threshold β.

4. Speaker segmentation via factor analysis

Our experiments indicate that the default LLR boundary generation of Section 2.1 frequently produces inaccurate boundaries. Furthermore in future setups we may have prior speaker information. This leads us to the idea to use the more advanced factor analysis based methods for speaker segmentation as well as these can exploit speaker-specific information more readily.

4.1. Factor analysis based boundary generation

In order to obtain accurate boundaries the decision to insert a boundary should happen after every frame (or very short block of frames). First, for each frame we extract speaker factors using a simple eigenvoice model [12].

$$\boldsymbol{m}_t = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{x}_t \quad (9)$$

The speaker factor extraction is based on the frames inside a window of length T centered around the considered frame at time t. Length T is identical to the enforced minimum duration of a speaker turn during peak selection of Section 2.1. $\boldsymbol{m}$ is the supervector of the UBM. Extractor matrix $\boldsymbol{V}$ contains the R eigenvoices obtained on the training data. We do not use the Total Variability framework as we want the speaker factors

to react on speaker changes only and not on intra-speaker variability. Thus, during the training we model the variability between speaker clusters (by pooling all speaker turns of the same speaker). In order to get a reasonable computational efficiency during evaluation the UBM has a low number of mixtures (=32) and matrix $\boldsymbol{V}$ is of low rank (=20).

Next, we look for significant local changes in the speaker factors which indicate a speaker change at time t. We therefore compare speaker factors at time $t-\tau$ and $t+\tau$. The time difference 2τ should not be significantly smaller than the extraction window length T as this leads to heavily overlapping analysis windows. 2τ should also not be too large, otherwise we may miss very short speaker turns. One option to compute the distance between speaker factors is the frequently used Cosine Distance Scoring (CDS) [2]

$$D_{\text{CDS}}(t) = 1 - \frac{\boldsymbol{x}_{t-\tau} \cdot \boldsymbol{x}_{t+\tau}}{\|\boldsymbol{x}_{t-\tau}\| \|\boldsymbol{x}_{t+\tau}\|} \tag{10}$$

Given the distances, the same peak selection criterion as described in Section 2.1 can be used to select likely speaker boundaries.

Another option for the distance measure is Euclidean distance

$$D_{\text{EUC}}(t) = \|\boldsymbol{x}_{t-\tau} - \boldsymbol{x}_{t+\tau}\| \tag{11}$$

Both distance measures listed above are not all that robust w.r.t. phonetic variability. Due to the short extraction window of 1s, the phonetic content has a huge impact on the value of $\boldsymbol{x}_t$. In [13] it is claimed that intra-speaker variability results in directional scattering of supervector $\boldsymbol{m}_t$. So the directions of $\boldsymbol{m}_t$ relative to the origin $\boldsymbol{m}$ deliver more speaker-specific information than the magnitudes. CDS exploits this fact via length normalization of $\boldsymbol{x}_t$. However, this procedure is sensitive to mismatches of the origin $\boldsymbol{m}$ between training and test data. A more robust way to compensate for the directional scattering may be the use of a Mahalanobis-based distance. We assume that the frames in a window of length $T_{\boldsymbol{\Sigma}}$ to the left of frame at time $t-\tau$ are uttered by the same speaker and we model the local phonetic variability with a Gaussian with mean $\boldsymbol{\mu}_L$ and full covariance matrix $\boldsymbol{\Sigma}_L$. Similarly we determine a $\boldsymbol{\Sigma}_R$ on the frames to the right of $t+\tau$. We can now define a distance measure as the sum of two Mahalanobis distances:

$$D_{\text{MAH}}(t) = \sqrt{\boldsymbol{x}_{t-\tau}^T \boldsymbol{\Sigma}_L^{-1} \boldsymbol{x}_{t+\tau}} + \sqrt{\boldsymbol{x}_{t-\tau}^T \boldsymbol{\Sigma}_R^{-1} \boldsymbol{x}_{t+\tau}} \tag{12}$$

This sum should get maximal when there are changes in x_t which do not get explained by changes in phonetic content, but rather by real speaker changes. Moreover, this approach should be much less sensitive to mismatches between training and test data since the phonetic variability $\boldsymbol{\Sigma}_{L(R)}$ is measured on the test data itself. Again, the peak selection remains unchanged.

4.2. Factor analysis for boundary elimination

In an initial set of experiments, we replaced the acoustic features in the ΔBIC criterion (3) with the speaker factors from (9). This however did not yield good results.

Since the factor analysis based boundary generation of Section 4.1 generates fewer false positives (see later), the average length of the segments is larger which in turn reduces the impact of phonetic variability when extracting speaker factors per segment. If sufficiently robust speaker factors can be extracted, the CDS can be used to eliminate boundaries as well. This option was tested with the same eigenvoice model as in Section 4.1. The elimination is stopped when the minimum CDS value exceeds a threshold α.

4.3. Two-pass speaker segmentation

The eigenvoices are determined on training data which may not really match with the evaluation data. In combination with the fact that we use low-dimensional models for computational reasons, this could result in degraded speaker segmentation models. This model mismatch can be eliminated in a two-pass system, since we can now use the speaker cluster output of an initial stage to retrain the eigenvoice model $\boldsymbol{V}$ per file. The UBM is retrained on the speech frames of the analyzed file as well. This two-pass approach should make the speaker factors much more robust against phonetic variability as the eigenvoices now form an exact match with the speakers in the file. The rank of $\boldsymbol{V}$ is limited to either the number of speakers in the file or R (the number of eigenvoices used in the first pass), whichever is the lowest. In the second pass, the whole file is resegmented with these new models.

5. Experiments

5.1. Data

All models are trained on 66 hours of speech from the 1996 HUB4 Broadcast News training data (3748 speakers). The evaluation corpus is the multilingual COST278 corpus[1]. It consists of complete TV news shows broadcasted by 16 European TV stations. It covers 9 national and 2 regional languages. Consult [4] and the website for more details. The corpus is divided into 12 language sets (but there are two Slovenian sets) of about three hours each. We used the BE language set for parameter tuning and the 11 remaining sets for evaluation. The evaluation data contains a total of 4386 speaker boundaries.

5.2. Evaluation measures

For the evaluation of the speaker segmentation the real (correct) and computed speaker change points are linked to one-another if the gap between both is not larger than a forgiveness collar of 500ms. The formed links determine the *recall* (percentage of real boundaries mapped to a computed one) and *precision* (percentage of computed boundaries mapped to a real one).

The Diarization Error Rate [14] is a popular metric to evaluate the performance of diarization systems. As all systems use the same oracle speech/non-speech marks we only study the relevant Speaker Error Rate (SER) component. This SER is the percentage of frames that are attributed to a wrong speaker given an optimal mapping between the speaker clusters and the reference annotation.

5.3. Boundary generation

We evaluate all boundary generation methods and compare them at different operating points using ΔBIC boundary elimination. The LLR boundary generation uses the following parameter settings: an LLR window size N_w of 200 frames (2s), averaging to eliminate spurious peaks is done across a window of 75 frames, a presumed speaker turn duration r of 5s and a minimum number of peaks to detect $N_{p,min}$ per speech segment of 3. We enforce a minimum duration of 1s for each speaker turn. The precision-recall PR curve in Figure 1 shows the performance of the boundary detection in function of the ΔBIC boundary elimination parameter λ. We notice the precision-recall trade-off for varying values of λ. The maximum recall in a realistic working point ($\lambda = 1.5$) is 76.8%.

[1] http://dssp.elis.ugent.be/cost278bn

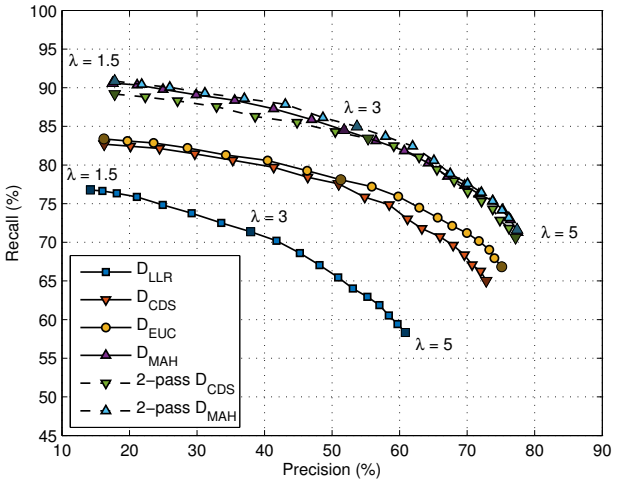

Figure 1: *Precision-recall curves for all boundary generation systems in function of the ΔBIC boundary elimination threshold λ.*

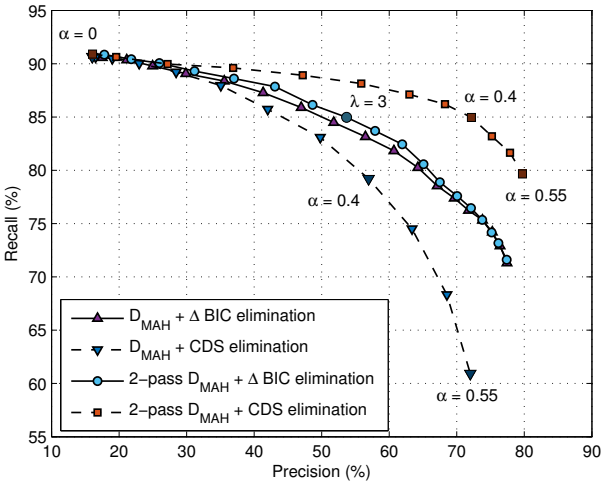

Figure 2: *Precision-recall curves for CDS boundary elimination vs. ΔBIC elimination in combination with the top-performing boundary generation methods in function of the thresholds λ, α.*

The boundary generation proposed in Section 4 uses a 32 mixture UBM. The rank R of $\boldsymbol{V}$ is set to 20. The speaker factor extraction uses a window size T of 100 frames (1s). Time difference τ is set to 25 frames which results in an overlap of 50% for the windows of $\boldsymbol{x}_{t-\tau}$ and $\boldsymbol{x}_{t+\tau}$. The distance measure D_{MAH} uses a window $T_{\boldsymbol{\Sigma}}$ of 175 frames to estimate the covariances $\boldsymbol{\Sigma}_L$ and $\boldsymbol{\Sigma}_R$. All other parameter settings remain identical to ones used in the LLR boundary generation.

All factor analysis based methods presented in Figure 1 clearly outperform the baseline LLR boundary generation. The use of distance measures D_{CDS} and D_{EUC} results in similar performance. D_{MAH} yields a maximum recall of 90.6% which is significantly better than all other methods.

5.4. CDS boundary elimination

We study the top-performing D_{MAH} boundary generation followed by either ΔBIC or CDS elimination. The PR curves can be found in Figure 2. CDS elimination is clearly outperformed by our default ΔBIC elimination.

5.5. Two-pass systems

The first pass always uses D_{MAH} with ΔBIC elimination. The parameter settings for clustering can be found in Section 5.6. In Figure 1 the two-pass D_{CDS} boundary generation achieves very similar results to two-pass D_{MAH}, which indicates the eigenvoices have become much more robust against phonetic variability. Adapting D_{MAH} does not yield huge improvements as this system was already quite robust. Figure 2 shows that matching eigenvoices allow us to exploit the full potential of the CDS boundary elimination as the the two-pass D_{MAH} with CDS elimination clearly outperforms all previous systems.

5.6. Clustering results

The iVector PLDA clustering uses an UBM of 256 mixtures and the rank of $\boldsymbol{T}$ and $\boldsymbol{V}$ is set to 100 and 80 respectively. We include extra information of the signal dynamics by including Δ-features in the feature vector. The threshold λ of the initial BIC clustering is set to 4.5 and the PLDA threshold β equals 2.5. The clustering starts from the segmentation results with settings $\lambda = 3.0$ or $\alpha = 0.4$. Clustering performance is analyzed with and without standard Viterbi resegmentation on acoustic MFCC features. The results can be found in Table 1.

resegmentation	no			yes		
	SER	P	R	SER	P	R
D_{LLR}+ΔBIC	11.2	53.8	69.6	10.7	62.1	76.0
D_{MAH}+ΔBIC	10.2	72.4	82.7	10.1	72.0	79.9
2-pass D_{MAH}+CDS	9.8	76.3	84.0	9.8	76.1	79.1

Table 1: Clustering performance (Speaker Error Rate, boundary Precision and Recall) with different speaker segmentation modules.

Our proposed factor analysis based speaker segmentation clearly results in more accurate boundaries. The two-pass system reduces the error rate on precision (P) and recall (R) by almost 50% relatively. After Viterbi resegmentation the gains are less pronounced. Whereas resegmentation improves boundaries generated with D_{LLR}, it deteriorates those generated by D_{MAH} slightly. The SER improves by 8% relatively from 10.7% to 9.8%.

6. Conclusions

We presented a factor analysis based speaker change detection that compensates for phonetic variability by using a Mahalanobis distance between speaker factors. The method reduces boundary detection errors by 50% relatively compared to a BIC baseline. The effectiveness of a two-pass strategy also indicates that the new method paves the way for new methods to exploit prior information given about speaker identities.

7. Acknowledgments

This research is sponsored by IWT Innovatief Aanbesteden within the scope of the VRT STON (Subtitling by using speech and language technology) project.

8. References

[1] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 127–132.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[3] J. Silovský, J. Prazak, P. Cerva, J. Zdánský, and J. Nouza, "PLDA-based clustering for speaker diarization of broadcast streams." in *Proc. Interspeech*, 2011, pp. 2909–2912.

[4] A. Vandecatseye, J.-P. Martens, J. Neto, H. Meinedo, C. Garcia-Mateo, J. Dieguez, F. Mihelic, J. Zibert, J. Nouza, P. David, M. Pleva, A. Cizmar, H. Papageorgiou, and C. Alexandris, "The COST278 pan-European broadcast news database," in *Proc. LREC*, 2004, pp. 873–876.

[5] B. Desplanques, K. Demuynck, and J.-P. Martens, "Combining Joint Factor Analysis and iVectors for robust language recognition," in *Proceedings of Odyssey 2014: The Speaker and Language Recognition Workshop*, 2014, 73–80.

[6] A. Vandecatseye and J.-P. Martens, "A fast, accurate and stream-based speaker segmentation and clustering algorithm," in *Proc. Eurospeech*, 2003, pp. 941–944.

[7] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proceedings of 2001: A Speaker Odyssey, The Speaker Recognition Workshop*, 2001, pp. 213–218.

[8] O. Glembek, L. Burget, P. Matějka, M. Karafiát, and P. Kenny, "Simplification and optimization of i-vector extraction," in *ICASSP*, 2011, pp. 4516–4519.

[9] L. Burget, P. Matějka, P. Schwarz, O. Glembek, and J. Černocký, "Analysis of feature extraction and channel compensation in GMM speaker recognition system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 1979–1986, 2007.

[10] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey 2010: The Speaker and Language Recognition Workshop*, 2010, p. 14.

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

[12] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *Proc. ICASSP*, 2008, pp. 4133–4136.

[13] H. Tang, S. M. Chu, and T. S. Huang, "Generative model-based speaker clustering via mixture of von mises-fisher distributions," in *Proc. ICASSP*, 2009, pp. 4101–4104.

[14] NIST, *The 2009 (RT-09) rich transcription meeting recognition evaluation plan*, 2009, http://www.itl.nist.gov/iad/mig//tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf.