Adaptive Speaker Diarization of Broadcast News Based on Factor Analysis

Brecht Desplanques[a,*], Kris Demuynck[a], Jean-Pierre Martens[a]

[a] *Ghent University - imec, IDLab, Department of Electronics and Information Systems, Sint-Pietersnieuwstraat 41 ,B-9000 Ghent, Belgium*

Abstract

The introduction of factor analysis techniques in a speaker diarization system enhances its performance by facilitating the use of speaker specific information, by improving the suppression of nuisance factors such as phonetic content, and by facilitating various forms of adaptation. This paper describes a state-of-the-art iVector-based diarization system which employs factor analysis and adaptation on all levels. The diarization modules relevant for this work are: the speaker segmentation which searches for speaker boundaries and the speaker clustering which aims at grouping speech segments of the same speaker. The speaker segmentation relies on speaker factors which are extracted on a frame-by-frame basis using eigenvoices. We incorporate soft voice activity detection in this extraction process as the speaker change detection should be based on speaker information only and we want it to disregard the non-speech frames by applying speech posteriors. Potential speaker boundaries are inserted at positions where rapid changes in speaker factors are witnessed. By employing Mahalanobis distances, the effect of the phonetic content can be further reduced, which results in more accurate speaker boundaries. This iVector-based segmentation significantly outperforms more common segmentation methods based on the Bayesian Information Criterion (BIC) or speech activity marks. The speaker clustering employs two-step Agglomerative Hierarchical Clustering (AHC): after initial BIC clustering, the second cluster stage is realized by either an iVector Probabilistic Linear Discriminant Analysis (PLDA) system or Cosine Distance Scoring (CDS) of extracted speaker factors. The segmentation system is made adaptive on a file-by-file basis by iterating the diarization process using eigenvoice matrices adapted (unsupervised) on the output of the previous iteration. Assuming that for most use cases material similar to the recording in question is readily available, unsupervised domain adaptation of the speaker clustering is possible as well. We obtain this by expanding the eigenvoice matrix used during speaker factor extraction for the CDS clustering stage with a small set of new eigenvoices that, in combination with the initial generic eigenvoices, models the recurring speakers and acoustic conditions more accurately. Experiments on the COST278 multilingual broadcast news database show the generation of significantly more accurate speaker boundaries by using adaptive speaker segmentation which also results in more accurate clustering. The obtained speaker error rate (SER) can be further reduced by another 13% relative to 7.4% via domain adaptation of the CDS clustering.

Keywords: speaker diarization, speaker segmentation, iVector, domain adaptation, 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). Numerous applications can benefit from such information. In this work, the ultimate goal is the semi-automatic creation of subtitles. VRT, the public broadcaster of Flanders, wants to speed up this subtitling process by employing speech technology. VRT's subtitles are primarily aimed at the hard-of-hearing and deaf. Hence, the subtitles must be of a very high quality and all spoken language constructs must be correctly converted to compact written forms. As a result, full automation is not an option yet. The main idea is therefore to reduce the manual work by letting the human operator correct the output of an automatic system, rather than starting from scratch. A detailed description of the full subtitle creation work flow can be found in (Verwimp et al., 2016). The output of the speaker diarization is used to add informative color codes to the generated subtitles and to profit from speaker adapted models during automatic speech recognition (ASR). We also note that ASR benefits from the location of the speaker change points as it gives an indication to the language model that a new sentence should be started at that place.

Speaker diarization encompasses both speaker segmentation and speaker clustering. After an initial speech/non-speech detection stage, the segmentation stage splits the continuous speech segments of the audio stream into homogenous segments with one active speaker. Next, the clustering stage groups the generated segments into clusters representing single speakers. There exist several approaches to speaker diarization such as Hierarchical Dirichlet Process Hidden Markov Modeling (HDP-HMM) (Fox et al., 2011), iterative mean shift clustering (Senoussaoui et al., 2014), K-means clustering (Shum et al., 2011), spectral clustering (Ning et al., 2006), Variational Bayes Expectation Maximization Gaussian Mixture Modeling (VBEM-GMM) (Shum et al., 2012), etc. In this paper we opt for bottom-up Agglomerative Hierarchical Clustering (AHC) that builds speaker models based on an initial speaker segmentation and successively merges the segments until one cluster per speaker remains. This AHC approach is by far the most popular technique and has proven to consistently achieve state-of-the-art results (Žibert et al., 2005; Zelenák et al., 2012; Bell et al., 2015).

We revise the initial segmentation stage because we noticed that inaccuracies in the boundaries can have a detrimental effect on both the speaker clustering and the speech recognition that follows. In previous work (Desplanques et al., 2015), we replaced our conventional speaker segmenter which employed log-likelihood ratios (LLR) and Bayesian Information Criterion (BIC) distances (Chen and Gopalakrishnan, 1998) in the acoustic feature space, by a segmenter that operates in the so-called speaker factor space. This resulted in enhanced boundary detection because phonetic variability is better suppressed in the speaker factor space, reducing the chance that phonetic variability is confused with a speaker transition. This state-of-the-art speaker factor extraction (SFE) method, which

*Corresponding author

Email addresses: brecht.desplanques@ugent.be (Brecht Desplanques), kris.demuynck@ugent.be (Kris Demuynck), martens@elis.ugent.be (Jean-Pierre Martens)

is described in Section 2.2.1, follows a paradigm that very much resembles the iVector paradigm proposed by Dehak et al. (2011). We propose to further enhance the SFE method by letting it differentiate between true speech frames and frames which belong to short silences between words or closures in plosives by including a soft voice activity detection (VAD) pre-processing step. This step generates frame-wise speech posteriors, and employs these probabilities to suppress the impact of the "nonspeech-like" frames on the speaker factors.

Although further improvements in speaker segmentation could be pursued by developing techniques that exploit prior speaker information retrievable from television show scripts, the improvements suggested here boil down to a better acoustic analysis that can also be applied if no script information is available. We will show that the transition to more speaker specific models results in substantial improvements for both speaker segmentation and speaker clustering. The factor analysis techniques use generic speaker models for both speaker segmentation an clustering as they should be able to discern between a wide range of speakers. However, after a first pass of the diarization algorithm more information about the active and relevant speakers is available. We propose to use this extra information to retrain the segmentation eigenvoice model. This ensures that the relevant speakers are modeled more accurately which in turn results in the generation of more accurate speaker boundaries during a second segmentation pass.

The actual clustering of the segments is performed by the two-step Agglomerative Hierarchical Clustering (AHC) system proposed in (Silovský et al., 2011). In this approach an initial BIC clustering stage is followed by iVector Probabilistic Linear Discriminant Analysis (PLDA) clustering. The iVector PLDA paradigm has shown to deliver state-of-the-art performance in the related field of speaker recognition (Bansé et al., 2014). In a way similar to the adaptive segmentation we can enhance the generic speaker clustering model. Given the subtitling use case, we can assume there are TV shows available that are related to the show in question. One option to use such additional information, is the longitudinal speaker diarization pursued in the recent 2015 Multi-Genre Broadcast (MGB) Challenge (Bell et al., 2015) where detected speakers were to be linked with previously broadcasted material of the same show. The approach proposed in this paper is more limited in scope but still allows us to enhance the speaker models based on speaker information of earlier episodes. This was achieved by replacing the iVector PLDA clustering by Cosine Distance Scoring (CDS) of speaker factors extracted by a generic eigenvoice matrix. Next, a more speaker specific set of eigenvoices based on earlier episodes is generated. Finally, to model recurring and important speakers more accurately the eigenvoice matrix is expanded with this extra set of eigenvoices and the CDS speaker clustering is re-executed.

The outline of this paper is as follows. The next section covers the use of proven factor analysis techniques to verify if snippets of speech are uttered by the same speaker or not. The following section explains how these techniques can be used at various stages of the speaker diarization in the context of automatic subtitling . Section 4 presents how the detected speaker change points can be made more accurate by incorporating voice activity detection and by exploiting speaker information in an unsupervised way. It also describes a similar way to apply domain adaptation to the speaker clustering. In the last section the proposed systems and adaptation techniques are evaluated on the COST278 multilingual broadcast news data set (Vandecatseye et al., 2004). We evaluate the boundary accuracy before and after clustering, discuss the speaker error rate, and

study the impact of the initial speech/non-speech detection on the speaker diarization.

2. Factor analysis based speaker characterization

A recurring problem in speaker diarization is the verification of the hypothesis that two sets of acoustic feature vectors $\mathcal{N}_i$ and $\mathcal{N}_j$ are uttered by one and the same speaker. In this work we will rely on two related factor analysis techniques that are predominant in the domain of speaker verification to extract speaker specific information and to evaluate this same-speaker hypothesis. The concepts of both approaches will be readily used throughout the remainder of this work.

2.1. Total Variability approach

The main idea behind the Total Variability (TV) (Dehak et al., 2011) approach is that there are different sources of variability between the acoustic frames in each set (speaker, channel, language, phonetic content, ...) and the emphasis during speaker characterization should be on the variability that is induced by changes of the speaker. The TV approach aims to describe this total acoustic variability in a low dimensional subspace. To that end, a fixed-length iVector is extracted for each set. The iVector includes information about all sources of variability but should be independent of the phonetic content encountered in the set. During the Probabilistic Linear Discriminant Analysis (PLDA) (Kenny, 2010) scoring stage this compact representation is used to extract and compare the speaker-specific information.

2.1.1. iVector extraction

The compact representations are estimated from a common Gaussian Mixture Model (GMM), called the Universal Background Model (UBM) for speech. First, a high-dimensional supervector $\boldsymbol{m}_{\text{UBM}}$ is constructed by concatenating all the mean vectors of the Gaussians in the speech UBM. Subsequently, to extract the iVector of set $\mathcal{N}$, a low rank matrix $\boldsymbol{T}$, called the TV matrix or the iVector extractor, is used to approximate the GMM mean supervector $\boldsymbol{m}_{\mathcal{N}}$ as

$$\boldsymbol{m}_{\mathcal{N}} = \boldsymbol{m}_{\text{UBM}} + \boldsymbol{T}\boldsymbol{x}_{\mathcal{N}} \tag{1}$$

where $\boldsymbol{x}_{\mathcal{N}}$ is the fixed length iVector that encodes the observed acoustics in a compact form. The prior distribution of the iVectors is assumed to be a standard normal distribution. By focusing on the shifts of the supervectors, the extracted iVectors should be largely independent of the phonetic content in set $\mathcal{N}$. The TV matrix is obtained on a large data corpus by means of Principal Component Analysis (PCA) initialization (Burget et al., 2007) followed by a number of iterations of the non-simplified Expectation-Maximization algorithm described by Glembek et al. (2011). The data corpus should contain a sufficient number of speakers that appear in multiple recordings. By treating all speaker turns as separate entities, the TV matrix is forced to learn both the within speaker variability and the across speaker variability.

2.1.2. PLDA scoring

Once iVectors of a sufficiently small dimensionality are obtained, a modified PLDA framework (Kenny, 2010) is used to highlight the speaker-specific components. In a first step, whitening and length normalization is applied to make the iVectors more Gaussian distributed (Garcia-Romero and Espy-Wilson, 2011). Then, each normalized iVector $\boldsymbol{x}_{\mathcal{N}}$ is modeled as

$$\boldsymbol{x}_{\mathcal{N}} = \boldsymbol{\mu} + \boldsymbol{V}_{\text{PLDA}}\boldsymbol{y}_{\mathcal{N}} + \boldsymbol{\epsilon}_r \tag{2}$$

where $\boldsymbol{\mu}$ is a global offset, $\boldsymbol{V}_{\text{PLDA}}$ represents the basis of the speaker-specific subspace and $\boldsymbol{y}_{\mathcal{N}}$ is a MAP point estimate of the latent variable $\boldsymbol{y}$ which is supposed to have a standard normal distribution. The residual term $\boldsymbol{\epsilon}_r$ is the nuisance variable which is computed with a zero-mean Gaussian with a full covariance matrix $\boldsymbol{\Sigma}_r$.

The same-speaker hypothesis can now be verified by estimating log-likelihood ratio

$$\text{LLR}_{\text{PLDA}}(\mathcal{N}_i, \mathcal{N}_j) = \log \frac{p(\boldsymbol{x}_{\mathcal{N}_i}, \boldsymbol{x}_{\mathcal{N}_j}|\mathcal{H}_s)}{p(\boldsymbol{x}_{\mathcal{N}_i}|\mathcal{H}_d)p(\boldsymbol{x}_{\mathcal{N}_j}|\mathcal{H}_d)} \tag{3}$$

where $\mathcal{H}_s$ is the hypothesis that the speech in $\mathcal{N}_i$ and $\mathcal{N}_j$ is uttered by the same speaker, $\mathcal{H}_d$ assumes different speakers. A closed-form solution of the log-likelihood ratio can be found in (Garcia-Romero and Espy-Wilson, 2011). The higher the ratio, the higher the likelihood of the same-speaker hypothesis.

2.2. Eigenvoice approach

Instead of delaying the extraction of the speaker-specific information to the scoring stage, one can extract the speaker-specific information during the estimation of the fixed-length representation of the acoustic feature vector set. This will enable the use of a more basic Cosine Distance Scoring (CDS) stage as the nuisance variability is already largely suppressed. We call this method the eigenvoice approach (Castaldo et al., 2008). This more basic approach allows for straightforward adaption strategies as will be explained in Section 4. The technique might also be more suited to extract low-dimensional speaker specific information from a very small acoustic feature set $\mathcal{N}$ as it immediately imposes the relevant constraints whereas the iVector approach has to work with a higher dimensional intermediate representation of all information. The sliding window speaker segmentation algorithm described in Section 3.3.1 certainly falls into this problem category.

2.2.1. speaker factor extraction

The procedure for extracting the speaker factors is similar to the iVector extraction described in Section 2.1.1. We also use the same training procedure to construct an eigenvoice matrix $\boldsymbol{V}$ instead of TV matrix $\boldsymbol{T}$. However, we want the speaker factors to react to speaker changes only and not to intra-speaker variability due to changes in the channel or the background. Thus, during the training of the eigenvoice matrix $\boldsymbol{V}$ we pool together all turns of a certain speaker into one instance of that speaker, meaning that the channel and background variability are incorporated in the speaker model.

2.2.2. CDS scoring

Tang et al. (2009) claim that intra-speaker variability results in directional scattering of the corresponding supervectors. So the directions of these supervectors relative to the UBM supervector deliver more speaker-specific information than the magnitudes. Cosine Distance Scoring (CDS) exploits this fact via length normalization of the extracted speaker factor vectors:

$$CDS(\mathcal{N}_i, \mathcal{N}_j) = 1 - \frac{\boldsymbol{x}_{\mathcal{N}_i} \cdot \boldsymbol{x}_{\mathcal{N}_j}}{\|\boldsymbol{x}_{\mathcal{N}_i}\| \|\boldsymbol{x}_{\mathcal{N}_j}\|} \tag{4}$$

The lower the distance score the higher the likelihood of the same-speaker hypothesis.

3. Speaker diarization for subtitling

The complete speaker diarization work flow intended for automatic subtitle creation is depicted in Fig. 1. The speaker diarization system is initialized by a speech/non-speech module which divides the audio into speech segments of at least 300ms long interleaved with long non-speech segments (having a length of at least 1 second). The speaker segmentation is then performed per speech segment whereas the speaker clustering considers all the speaker segments across speech segments in the entire audio file. The proposed segmentation into speaker turns is achieved by means of a two-stage procedure, as explained in (Desplanques et al., 2015; Vandecatseye and Martens, 2003). The first stage generates boundaries on the basis of a sliding window approach. We will refer to this process as the *boundary generation* stage. The second stage eliminates as many of the false positives as possible on the basis of similarities between adjacent segments of variable length as they emerge from the first stage, we call this *boundary elimination*. Then, two-stage agglomerative clustering is employed to group the detected speaker turns into speaker clusters representing single speakers. Recent evaluation on the 2015 Multi-Genre Broadcast MGB Challenge (Bell et al., 2015) of systems such as the one described in (Karanasou et al., 2015) has shown that this agglomerative clustering approach delivers state-of-the-art diarization results for broadcast media. Finally, an adaptive language identification strategy determines which speech recognizers to use in the subsequent stages (Verwimp et al., 2016).

Fig. 1 also includes adaptation schemes for both speaker segmentation and clustering. A two-pass approach allows for more precise speaker segmentation, whereas domain-adapted models help improve the speaker clustering. More details about this adaptation strategy can be found in Section 4. The following subsections will handle the baseline diarization modules in more detail.

3.1. Acoustic feature extraction

The diarization system (segmentation and clustering) works on 10ms frames. For each frame, 16 MFCCs (C_1-C_{16}) and a normalized log-energy are computed. The latter is defined as

$$\log E_{\text{nrm}}(t) = \log E(t) - \overline{\log E(t)} \tag{5}$$

It is equal to zero when the log-energy is equal to a running mean log-energy $\overline{\log E(t)}$ and positive when it is larger. The running mean is computed by means of a leaky integrator

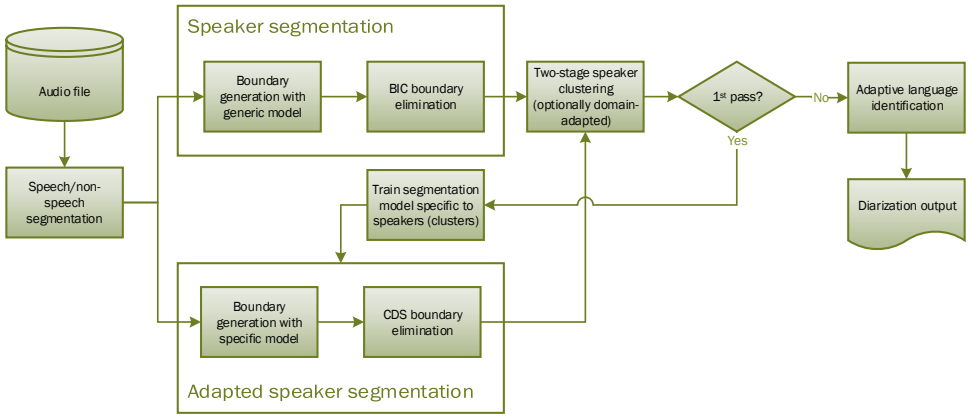

Figure 1: The complete speaker diarization process from audio file to annotations for automatic subtitling.

with a time constant of 5 seconds. A basic voice activity detection discarding all frames with a $\log E_{\text{nrm}}(t) < -1$ for example results in the removal of all frames with an energy level that is more than 4.3dB below the running mean energy, and was found to strike a good balance between retaining most of the speech frames and removing close to all inter-word silence frames. A more aggressive configuration with a positive threshold of 0.5 on the other hand only maintains frames with a relatively high energy level, most likely corresponding with syllable nuclei.

3.2. Speech/non-speech segmentation

The speech/non-speech module detects long non-speech intervals (>1s) that need no lexical transcription. These intervals can be discarded in the further processing of the audio stream. Non-speech intervals can contain music and strong background sounds such as applause and street noise, so we rely on a model-based approach (Desplanques and Martens, 2013) to detect these segments.

The sequential modeling is performed by means of a Hidden Markov Model (HMM) and the acoustic likelihoods are provided by Gaussian Mixture Models (GMMs) as shown in Fig. 2. The HMM comprises just one speech state (=S) and one non-speech (=NS) state and the sequence modeling is controlled by a transition penalty P_s. The S (or NS) state likelihood at some time is obtained by taking the maximum of the scores computed by a small set of GMMs representing specific acoustic conditions within the S (or NS) category. It utilizes the subcategories *broadband speech, telephone band speech, speech+music* and *speech+other* for state S and the subcategories *music* and *other* for state NS. To reduce the sensitivity of the model-based method to mismatches between the test and the training data, the acoustic models are MAP-adapted (Reynolds, 1997) per TV show in a two-pass approach. By increasing the penalty on transitions (lowering P_s) one increases the average duration of the S and NS segments being generated, but there is no absolute minimum segment duration. In order to achieve that, the outputs of the system are supplied to a post-processor which eliminates speech segments that are

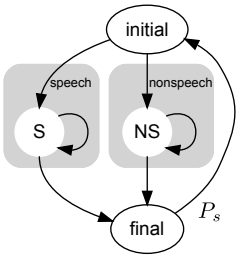

Figure 2: Speech/non-speech HMM.

shorter than 300ms and non-speech segments that are shorter than 1.0s. These duration thresholds are chosen to comply with the guidelines given to the annotators of the test data.

3.3. Speaker segmentation

This section describes the two-stage procedure that generates a set of potential speaker change points and then reduces it by local clustering.

3.3.1. Boundary generation

Each detected speech segment is analyzed for potential speaker changes. Candidate change points are generated at places of maximum difference between the statistical distributions of the acoustic vectors in two fixed-length windows to the left and to the right of the potential boundary. In (Desplanques et al., 2015) we proposed such a boundary generation method based on eigenvoices as a replacement for a conventional log-likehood ratio (LLR) based criterion. The online speaker factor extraction (SFE) produces speaker factors $\boldsymbol{x}_t$ (Section 2.2.1) at time t by considering the frames inside a window of length T_e centered around t (Castaldo et al., 2008). A detection of significant local changes in the speaker factors is then used to localize potential speaker changes. In order to constrain the computational load we use GMMs with a low number of mixtures (=32) and a low rank eigenvoice matrix $\boldsymbol{V}$ (rank = 20).

To assess the plausibility of having a speaker change at time t, we compare the speaker factors found at times $t-\tau$ and $t+\tau$ and we define $\Delta\boldsymbol{x}_t$ as $\Delta\boldsymbol{x}_t = \boldsymbol{x}_{t-\tau} - \boldsymbol{x}_{t+\tau}$. Sensible values for the time difference 2τ (in frames) are values close to or somewhat smaller than the window length T_e used for extracting the speaker factors. On the one hand, the time difference 2τ (in frames) should not be much smaller than the window length T_e as this would imply a significant overlap between the windows that give rise to the two speaker factors being involved. On the other hand, 2τ should also not be too large either, because we do not want to miss a short speaker turn that could be located in the gap between the two extraction windows.

Due to the rather small size of the speaker factor extraction window ($T_\mathrm{e} = 1.0$s), the phonetic content in the extraction window has a significant impact on the extracted $\boldsymbol{x}_t$. Let us now define a window of length $T_{\boldsymbol{\Sigma}}$ to the left of $t-\tau$ and assume that (a) the frames in that window stem from the same speaker and (b) the statistics of the speaker factors in that window are represented by a full-covariance Gaussian distribution with means $\boldsymbol{\mu}_{L,t}$ and covariances $\boldsymbol{\Sigma}_{L,t}$. Similarly the statistics of the speaker vectors in a

window of the same length to the right of $t + \tau$ yield means $\boldsymbol{\mu}_{R,t}$ and covariances $\boldsymbol{\Sigma}_{R,t}$. Under these hypotheses the covariance matrices $\boldsymbol{\Sigma}_L$ and $\boldsymbol{\Sigma}_R$ are expected to model the phonetic variability within speech of the left and right speaker respectively. The following distance (the sum of two Mahalanobis distances)

$$D_{\mathrm{MAH}}(t) = \sqrt{\Delta \boldsymbol{x}_t^T \boldsymbol{\Sigma}_{L,t}^{-1} \Delta \boldsymbol{x}_t} + \sqrt{\Delta \boldsymbol{x}_t^T \boldsymbol{\Sigma}_{R,t}^{-1} \Delta \boldsymbol{x}_t} \tag{6}$$

is then expected to reach a maximum when the changes in $\boldsymbol{x}_t$ cannot be explained by changes in the phonetic content alone. Moreover, since the phonetic variability $\boldsymbol{\Sigma}_{L(R)}$ is measured on the test data itself, the approach is presumed to be insensitive to mismatches between training and test data. This Mahalanobis metric outperforms the more basic Euclidean metric and cosine distance based scoring (Desplanques et al., 2015).

To avoid the detection of spurious peaks, $D_{\mathrm{MAH}}(t)$ is first smoothed by a moving average filter that uses a hamming window of length N_{avg}. For each speech segment $\mathcal{S}$ up to $N_{\mathrm{p}}(\mathcal{S})$ of the largest peaks are selected in the smoothed pattern. The number of peaks N_{p} is chosen proportional to the duration $T(\mathcal{S})$ of $\mathcal{S}$:

$$N_{\mathrm{p}}(\mathcal{S}) = \max(N_{\mathrm{p,min}}, \left\lceil \frac{T(\mathcal{S})}{T_{\mathrm{masl}}} \right\rceil) \tag{7}$$

$N_{\mathrm{p,min}}$ is the minimum number of peaks to detect and T_{masl} denotes the minimal average length of the speaker segments one wants to enforce (e.g. 5 seconds). The introduction of $N_{\mathrm{p,min}}$ allows for turn-taking within very short speech segments. However, these short speaker turns are rare in the broadcast news domain and the impact of this parameter negligible. In addition we prevent the system from generating speaker segments that are shorter than T_{min} (fixed to 1 second).

Note that this general peak detection algorithm can be readily combined with other distance measures that express the difference in statistical distributions of the acoustic vectors in the two sliding comparison windows. A frequently used metric in that regard is the log-likelihood ratio (LLR) verifying the hypothesis of having two speakers versus one speaker when the data in the comparison window are modeled with a single Gaussian per speaker:

$$D_{\mathrm{LLR}}(t) = 2 \log |\boldsymbol{\Sigma}_{L+R}| - \log |\boldsymbol{\Sigma}_L| - \log |\boldsymbol{\Sigma}_R| \tag{8}$$

with each $\boldsymbol{\Sigma}$ the Maximum Likelihood (ML) full covariance of the acoustic features in the left (L), right (R) and merged (L+R) window. A detailed analysis of the D_{LLR} metric is given by Desplanques et al. (2016).

The peak detection algorithm can also be used in combination with the normalized log-energy defined in Eq. (5). High values of $\log E_{\mathrm{nrm}}$ indicate speech activity. Pauses between words and sentences on the other hand correspond with troughs in the energy signal and can be used to insert potential speaker change points as there is a low chance of incorrectly splitting up words. The peak detection is applied on the smoothed $-\log E_{\mathrm{nrm}}$ signal in order to detect the relevant negative peaks.

3.3.2. Boundary elimination

The operating point of the boundary generation stage is set to maximize the recall at the cost of a lower precision. The hope is that by performing an agglomerative clustering of adjacent segments on the basis of the Bayesian Information Criterion (BIC) (Chen and

Gopalakrishnan, 1998), it will be possible to reach a working point that is well above the point with a similar recall/precision trade-off that could be reached with the boundary generation stage alone. The segmental BIC distance (Stafylakis et al., 2010) between two segments is given by

$$\begin{aligned}\Delta BIC = (N_L + N_R)\log|\mathbf{\Sigma}_{L+R}| \\ - N_L \log|\mathbf{\Sigma}_L| - N_R \log|\mathbf{\Sigma}_R| - \lambda P\end{aligned} \tag{9}$$

with N and $\mathbf{\Sigma}$ being the number of frames and the full covariance matrix of the feature vectors in the considered segment and with P being given by

$$P = \frac{1}{2}\left(d + \frac{1}{2}d(d+1)\right)\log\frac{N_L \times N_R}{N_L + N_R} \tag{10}$$

where d is the dimension of the feature vectors. Stafylakis et al. (2010) introduced segmental BIC as a replacement for the local BIC with $P \propto \log(N_L + N_R)$, in order to more accurately penalize the parameters of the models with the effective sample size. Starting from the segment set of the analyzed speech segment, an iterative procedure merges the two adjacent segments with the lowest ΔBIC for as long as this value is negative. Obviously, at every merge, the ΔBIC values of the endpoints of the newly formed segment have to be updated. The parameter λ in Eq. (9) controls the number of boundaries that will be eliminated.

3.4. Speaker clustering

The detected speaker segments are finally clustered using the two-stage Agglomerative Hierarchical Clustering (AHC) approach proposed by Silovský et al. (2011). Two-stage speaker clustering systems have proven their efficiency before (Zhu et al., 2005) and the main idea is to rely on more complex speaker identification methods as the clustering advances and the clusters contain more acoustic data on average. Each cluster in the final output is supposed to encompass all the segments of a particular speaker. The AHC algorithms do not fix the number of speakers explicitly but merge clusters based on a series of speaker recognition decisions.

The segments returned by the boundary elimination stage may still be rather short, and hence the first stage employs robust techniques such as ΔBIC that are known to work well even if only a limited amount of frames is available. As few as 30 frames per segment is possible, which corresponds with the imposed minimum duration of a speech segment. The agglomerative clustering starts with as many clusters as there are speaker segments and it gradually merges the two most similar clusters until the ΔBIC distance between these clusters, defined in Eq. (9), turns out to be positive.

In the second stage, the segments are longer on average and hence more advanced techniques can be used. First of all, the second stage clustering discards the frames with a low $\log E_{\mathrm{nrm}}$ because the spectral content of these frames (e.g. the closures in plosives) is frequently dominated by background noise. Second, the acoustic features of the selected frames are normalized by means of Feature Warping (Pelecanos and Sridharan, 2001). This technique transforms the individual features via a monotonic non-linear function so that their distribution over the processed time interval fits the standard normal distribution. The normalized features are more robust against additive noise and

channel mismatch. For this work, we apply Feature Warping on all the speech frames in the clusters returned by the first stage. Whereas the use of a sliding window approach is typically advocated as a means to limit the influence of changing noise/channel conditions, we observed that different noise/channel conditions give rise to different "first stage" clusters anyhow, alleviating the problem altogether. Third, iVector Probabilistic Linear Discriminant Analysis (PLDA) described in Section 2.1 is used to iteratively merge the BIC clusters on the basis of these normalized feature vectors.

When the two most similar clusters are being merged during PLDA clustering a new iVector has to be computed for the new cluster. This is realized by extracting a new iVector based on the summed up Baum-Welch statistics of the two merged clusters. The relevant log-likelihood ratios are updated and the clustering process is terminated when all ratios fall below a predetermined threshold β_{PLDA}.

We note that speaker recognition saw some recent gains by incorporating Deep Neural Network (DNN) senone posteriors as a replacement for the UBM posteriors into the iVector extraction (Kenny et al., 2014). But for now there are no indications yet that this approach will consistently result in significant improvements for the speaker diarization task (Sell et al., 2015, 2016).

3.5. Adaptive language identification

In Flanders foreign speech in international news items is not dubbed but presented with subtitles. As a result, language recognition (LR) is an indispensable pre-processor in any computer assisted TV captioning system for a Flemish TV broadcaster.

In (Desplanques et al., 2014), we assume that each speaker (cluster) uses one language only and depart from an acoustic language identification system that employs an iVector-based technique to characterize the speech. However, contrary to common practice, it does not project all variability in a single Total Variability space, but it utilizes Joint Factor Analysis (JFA) (Kenny et al., 2007) to separate the language variability from the speaker variability. In other words, it simultaneously extracts language factors as well as speaker factors per speaker. The language classification is implemented by a simple Gaussian back-end operating on the language factors. To cope with foreign accents and the everlasting influence of dialect on the standard language the language factors and the back-end are adapted to the language variants encountered in the audio file that is being processed. Experiments show reductions of speaker-based error rate by more than 20% relative by adapting the language variability model. We refer to (Desplanques et al., 2014) for more details.

4. Proposed adaptive framework for speaker diarization

In this section we propose methods to integrate our factor analysis based speaker diarization into an adaptive framework. The challenges are again tackled with an eye to better and more robust automatic subtitle generation.

4.1. Adaptive speaker segmentation

The eigenvoice model of Section 3.3.1 may be trained on data which may deviate substantially from the evaluation data. This, in combination with the fact that we use

low-dimensional models for computational reasons, can result in sub-optimal speaker segmentation models. The model mismatch can be eliminated by making the segmentation system adaptive in a two-pass approach. The first and second iteration employ the same SFE segmentation algorithm but with different UBMs and different eigenvoices. There are also arguments for choosing a different boundary elimination criterion after adaptation. Furthermore, we integrate a soft voice activity module in this adaptive speaker segmentation framework so that the speaker change detection is based on speaker information only. The adaptive work flow is shown in Fig. 1.

4.1.1. Adaptive soft voice activity detection

We incorporate a soft Voice Activity Detector (VAD) in the speaker factor extraction of the speaker segmentation to make it differentiate between speech frames and non-speech frames, as the non-speech frames are not expected to contribute information concerning the speaker identity. We chose to implement the soft VAD using a simple GMM-based approach (McLaren et al., 2015). This involves the training of a speech UBM θ_S and a non-speech UBM θ_{NS} on some training data. The speech GMM is trained on the high-energy frames ($\log E_{\text{nrm}} > -1$) found in the speech segments. The non-speech training data is created by pooling the low-energy speech frames and the non-speech frames. Note that the introduction of a voice activity detection (VAD) has already become common practice in related fields such as speaker recognition and language recognition. See e.g. (Ferrer et al., 2013, 2015).

The frame-wise speech posteriors generated by the VAD are used to weigh the detected speech frames during the speaker factor extraction. This modifies the intermediate estimation of the zero- and first-order Baum-Welch statistics (Glembek et al., 2011) to

$$N_{\mathcal{X}}^{m} = \sum_{\boldsymbol{o}_t \in \mathcal{X}} p(\theta_S|\boldsymbol{o}_t)\, \gamma(\theta_{S,m}|\boldsymbol{o}_t) \tag{11}$$

$$\boldsymbol{f}_{\mathcal{X}}^{m} = \sum_{\boldsymbol{o}_t \in \mathcal{X}} p(\theta_S|\boldsymbol{o}_t)\, \gamma(\theta_{S,m}|\boldsymbol{o}_t)\, \boldsymbol{o}_t \tag{12}$$

with $\gamma(\theta_{S,m}|\boldsymbol{o}_t)$ being the occupation probability of mixture m of the speech GMM and $\mathcal{X}$ being the relevant set of frames for which the speaker factors are extracted. The speech posterior $p(\theta_S|\boldsymbol{o}_t)$ for observation (frame) $\boldsymbol{o}_t$ is obtained by combining the speech and non-speech log-likelihoods $\log p(\boldsymbol{o}_t|\theta_{S/NS})$ with their priors $P_{S/NS}$ as follows:

$$p(\theta_S|\boldsymbol{o}_t) = \frac{p_S\, e^{\rho \log p(\boldsymbol{o}_t|\theta_S)}}{p_S\, e^{\rho \log p(\boldsymbol{o}_t|\theta_S)} + p_{NS}\, e^{\rho \log p(\boldsymbol{o}_t|\theta_{NS})}} \tag{13}$$

Factor ρ can be manipulated to calibrate the speech posteriors. For our application the impact of ρ is rather limited and it is therefore fixed to $\rho = 1.0$. We also assume equal priors for both classes in all experiments.

To optimize the soft VAD in the two-pass system, we retrained both the speech GMM and the non-speech GMM in the second pass. Similar to the training of the default models we retrain the speech GMM on the high-energy frames in the speech regions of the file and the NS model on the low-energy frames in the speech regions as well as on the frames in the non-speech regions of the file. The weighted Baum-Welch statistics needed for the eigenvoice model retraining described in the next section on the other hand are extracted across all frames of the speech regions belonging to the considered speaker cluster.

4.1.2. Eigenvoice model retraining

We use the speaker clusters emerging from the first pass and the retrained GMMs to create a new eigenvoice model $\boldsymbol{V}$ for the file under analysis and we repeat the segmentation with the new models. As the eigenvoices now match the speakers in the file well, the speaker factors are expected to be more robust against phonetic variability. The rank of the retrained eigenvoice matrix $\boldsymbol{V}$ is either the same as that of the original matrix, or it is changed to the number of clusters emerging from the first pass, whichever is the lowest.

4.1.3. CDS boundary elimination

In (Desplanques et al., 2015) we showed that Cosine Distance Scoring (CDS) outperforms BIC in the boundary elimination stage of a speaker segmenter when the eigenvoices match the test data well. For each speaker segment s inside a speech segment $\mathcal{S}$ we extract speaker factors $\boldsymbol{x}_s$ using the new eigenvoice model and we merge the adjacent segments exhibiting the lowest cosine distance. The elimination in the second iteration continues until the lowest CDS value exceeds a predefined threshold α_{CDS}.

4.2. Domain adaptation for speaker clustering

In our subtitling use case we can assume that there is a small amount of TV shows available that are related to the show in question. Given our successful attempts to exploit speaker specific information during speaker segmentation we argue that more speaker specific models can also benefit the speaker clustering stage. The main idea is that by dedicating more parameters of the speaker models to relevant and recurring speakers one can significantly improve the speaker clustering. We will refer to this adaptation process as *domain adaptation*. Note that initial attempts to only exploit speaker information inside the considered audio file (similar to the adaptation of the speaker segmentation) were unsuccessful and only led to an reinforcement of errors made in the first diarization pass.

Adaptation of the PLDA parameters has been successfully explored before in speaker recognition tasks (Garcia-Romero and McCree, 2014). The basic idea is to adapt an existing resource-rich out-of-domain system by using a small amount of in-domain data. This could for example be achieved by interpolating between an in-domain PLDA model and an out-of-domain PLDA model. Even unsupervised adaptation in which the in-domain speaker labels were automatically generated still resulted in significant gains (Brummer et al., 2014). Unsupervised PLDA adaptation was also beneficial for cross-show speaker diarization (Le Lan et al., 2016) when the complete target corpus with a few hundreds of speakers was used as adaptation data. However, in our system, we also want to cover the scenario were only limited amounts of unlabeled in-domain data with few speakers and few sessions per speaker are available. In these conditions, adaptation towards an in-domain PLDA model is expected to be problematic. Therefore, we will fall back on a more simple but more robust adaptation technique.

4.2.1. CDS eigenvoice clustering

In order to avoid the complex PLDA adaptation, the combination of iVectors and PLDA scoring (Total Variability modeling followed by a speaker sensitive distance metric) is replaced by Cosine Distance Scores (CDS) on speaker factors extracted by an eigenvoice matrix (i.e. more selective features combined with a more generic distance metric).

Although this could result in a small degradation of the diarization accuracy (Sell and Garcia-Romero, 2014), the adaptive speaker segmentation already learned us that CDS clustering is highly effective when the eigenvoices closely match the test data.

For the CDS-based clustering, we apply the same pre-processing as before: energy-based frame selection and feature normalization by means of Feature Warping. For each cluster a speaker factor factor is extracted through an eigenvoices matrix $\boldsymbol{V}$ which models the speaker variability in a low dimensional subspace. Details concerning the training of this eigenvoice matrix are given in the next section. The speaker factors are expected to reflect the speaker's characteristics in a compact form, and will serve as input for the CDS. The cluster algorithm merges the cluster pair with the lowest distance first. After each merge operation, new speaker factors are extracted for the resulting "merged" cluster. The cluster algorithm stops when all cluster pair scores exceed the threshold β_{CDS}.

4.2.2. Expanding the eigenvoice model

To train the UBM and the generic eigenvoice model V_{generic} that covers a broad range of speakers we use a large but out-of-domain database containing labeled data. The procedure is identical as described in Section 2.2.1. Thus, during the training we pool together all speech of a certain speaker. The rank R_{V_g} of the eigenvoice matrix V_{generic} is set to 100 for our experiments.

For each TV show we can easily obtain some in-domain data by selecting related broadcasts. This in-domain data contains both speech from key speakers appearing in several episodes and speech uttered by guest speakers appearing only once. All data is considered useful since our adaptation scheme primarily wants to learn the characteristics typical to the specific TV show such as live versus prepared audio, languages or dialects being spoken, recording and background conditions, etc. Note that by being more prominently present, key speakers will be better represented in the eigenvoices and hence the average accuracy on the key speakers will automatically improve. In case the adaptation data is unlabeled, we use the generic speaker cluster model to generate speaker labels. Once the labels are acquired one can train an in-domain eigenvoice model V_{specific}. The rank R_{V_s} of this eigenvoice matrix is either limited to R_{V_g} or the number of speakers in the adaptation data, whichever is the lowest. Finally, both sets of eigenvoices are concatenated: $\boldsymbol{V} = [V_{\text{generic}}, V_{\text{specific}}]$, hence combining the specificity of the in-domain eigenvoices with the robustness and broad coverage of the generic eigenvoices. The adaptation process keeps the original UBM unchanged, so that the original V_{generic} remains meaningful. Note that preliminary experiments with more complex adaptation schemes which trained a completely new eigenvoice model on the original training data pooled with the adaptation data did not deliver similar improvements, even when the adaptation data was weighed more heavily during the training.

One could argue that in the unsupervised adaptation case (no speaker labels available), speakers should be linked across TV shows in the in-domain data set. For the current setup the difference between the number of detected speakers and the effective number of speakers in the in-domain data is small and the impact is negligible as shown by our experimental results. But in the case of e.g. call center data with fixed operators the number of estimated eigenvoices could get significantly larger than the real number of speakers and speaker linking across telephone conversations might be recommended.

5. Experimental conditions

5.1. Training data

The English 1996 HUB4 Broadcast News (Garofolo et al., 1997) training data (66 hours, 3009 speakers) is the main source for training the various models. This includes the speech GMM, non-speech GMM and eigenvoice model $\boldsymbol{V}$ needed for the speaker segmentation. For the clustering, we either train an UBM, Total Variability matrix, whitening matrix and PLDA model (iVector+PLDA clustering) or an UBM and generic eigenvoice matrix (CDS clustering on speaker factors). The initial speech/non-speech segmentation uses a wider variety of speech and non-speech data sources including music and telephone data. For a full description of that data, we refer to (Desplanques and Martens, 2013).

5.2. Evaluation data and experimental setup

The evaluation corpus is the multilingual COST278 corpus[1]. It is composed of complete TV news show broadcasts by 16 European TV stations. It covers 9 national and 2 regional languages (Basque, Croatian, Czech, Dutch, Galician, Greek, Hungarian, Portuguese, Slovakian, Slovenian, Spanish). The corpus is divided into 12 sets of about three hours each: one set per language except for the 6 hours of Slovenian which is divided into two sets. Each hour of audio contains 55 minutes of speech on average.

The Belgian Dutch (BE) language set was set aside for parameter tuning and the 11 remaining language sets were used for evaluation. The evaluation data consists of 5997 speaker segments uttered by 2286 speakers. There are 4569 speaker changes within the continuous speech segments. Please consult (Vandecatseye et al., 2004) and the website for more details about the corpus.

5.3. 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. The error margin is set to either 500ms or 1s, depending on the goal of the performance analysis. 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 (NIST, 2009) is a popular metric to evaluate the performance of diarization systems. In the case that the systems are initialized by 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. If automatic speech/non-speech is enabled we fall back to the original Diarization Error Rate (DER). This DER is defined as the sum of a False Alarm (FA) component (percentage of non-speech detected as speech), Missed Speech (MS) (percentage of speech detected as non-speech) and the SER, now relative to the total duration of the file.

Note that with a final goal of building a semi-automatic subtitling tool we slightly deviate from the NIST standard formulation of the error rates. All error components of

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

the DER are relative to the total duration of the file instead of the cumulative duration of speech as we want the DER to correlate better with the total time it will take to correct the output. For example, a nature documentary can include lots of non-speech sounds (music, noises,...), and the FA component could undesirably become too dominant in the NIST formulation of the DER. We also do not allow a forgiveness collar around the real speaker and speech/non-speech changes as we want the DER to directly reflect if words could be dropped or assigned to the wrong speaker in the subsequent subtitle generation process. However, the annotation protocol specified that at the top level the audio should be split into speech segments separated by non-speech segments with a duration of at least 1 second. Shorter non-speech segments should be detected as speech and are included in the evaluation. The rationale was that this should be achievable by automatic speech/non-speech segmentation (Desplanques and Martens, 2013) and that subsequent processing of the speech segments (e.g. speech recognition) can be made robust against these short pauses between words and sentences. Finally, we do not include overlapping speech in our evaluation as this is not specifically annotated in the COST278 evaluation data set.

6. Experimental results

In order to separate the contribution of the various components and techniques, we first evaluate the performance of the techniques when starting from an oracle speech/non-speech (SNS) annotation. In Section 6.3 we will verify if the conclusions still hold when the diarization system is initialized with the output of an automatic SNS segmentation system.

6.1. Speaker segmentation

First, we discuss the performance of the baseline segmentation systems described in Section 3.3 and continue with the proposed adaptation framework proposed in Section 4.1.

6.1.1. Baseline speaker segmentation

The speaker segmentation is performed independently for each of the given speech segments. We compare the speaker segmentation performances of three baseline systems: (1) minimal $\log E_{\text{nrm}}$ + BIC-based boundary elimination, (2) baseline LLR boundary generation + BIC-based boundary elimination and (3) SFE boundary generation + BIC-based boundary elimination. The boundary generation of the first system looks for pauses between words and inserts potential speaker change points at places where the $\log E_{\text{nrm}}$ is minimal. The LLR boundary generation of the second system uses overlapping comparison windows to enhance the accuracy of the detected boundaries. More details and optimal parameters settings of this LLR system can be found in (Desplanques et al., 2016). We note that the SFE system uses the soft VAD speaker factor extraction proposed in Section 4.1.1. This did not result in significant performance gains in this one-pass approach.

All speaker segmentation parameters are tuned to get optimal precision-recall (PR) curves on the development COST278 BE data. The minimum number of selected peaks per speech segment $N_{\text{p,min}}$ is fixed to 3 (if there are that many peaks). We enforce a

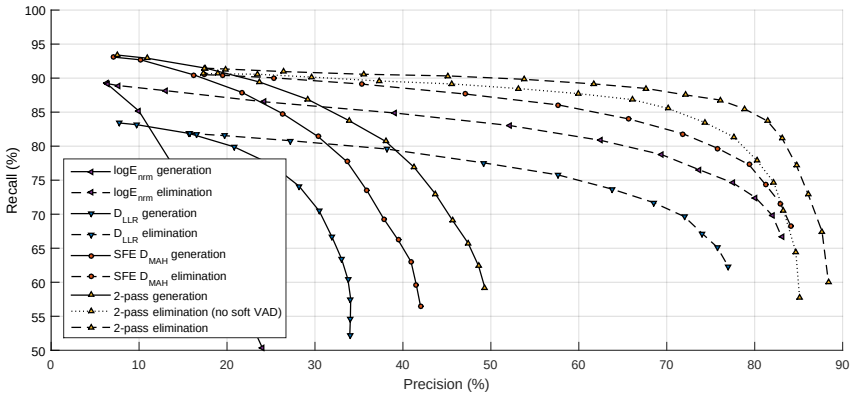

Figure 3: Precision-recall curves of the proposed speaker segmentation methods on the COST278 test data. The error margin is set to 500ms, and hence focus is on the accuracy of the position of the estimated speaker change points.

minimum average length of the generated speaker segments T_{masl} of 5 seconds unless mentioned otherwise and the minimum speaker turn duration T_{min} is set to 1 second. The moving average window length N_{avg} of all systems is fixed to 150 frames. The SFE boundary generation parameters are: the number of mixtures of the speech/non-speech GMM (32), the rank R of $\boldsymbol{V}$ (20), the speaker factor extraction window size T_{e} (1.0s), the time difference τ (250ms), and the window size $T_{\boldsymbol{\Sigma}}$ used for estimating $\boldsymbol{\Sigma}_L$ and $\boldsymbol{\Sigma}_R$ in Eq. (6) (1750ms). All parameters have sensible values given their function described in Section 3.3.1 and deliver consistent performance across all language sets of the COST278 data set. Note that preliminary experiments on the AMI Meeting Corpus Carletta et al. (2006) revealed that lower values for T_{masl}, $T_{\boldsymbol{\Sigma}}$ and N_{avg} are recommended for meeting data or other domains in which short speaker turns are commonplace, with T_{masl} being the most critical parameter.

The boundary generation (without boundary elimination) performance for all approaches is evaluated in combination with the peak detection algorithm described in Section 3.3.1. The parameter T_{masl} in Eq. (7) is used to create the precision-recall (PR) curves of the systems. The PR curves of the test data are generated for error margins of 500ms (Fig. 3) and 1000ms (Fig. 4). The more strict 500ms margin curves better reflect how accurate the positions of the boundaries are, whereas the broad 1000ms margin curves reveal how many speaker changes are actually detected. The initial and final points of the speech fragments (speech/non-speech boundaries) are excluded from the evaluation because they would always turn out to be correct in our experiment.

Surprisingly, the relatively simple $\log E_{\text{nrm}}$ boundary generation outperforms the more complex LLR method for low precision values. This corresponds with low values of T_{masl} and the generation of many short segments. The performance of $\log E_{\text{nrm}}$ quickly drops as T_{masl} increases. This can be explained by the fact that low values of $\log E_{\text{nrm}}$ coincide with pauses between words including speaker changes, but there is no reason to assume that pauses between speakers correspond with lower values compared to the inter-word pauses. Thus potential speaker changes will be incorrectly dropped as

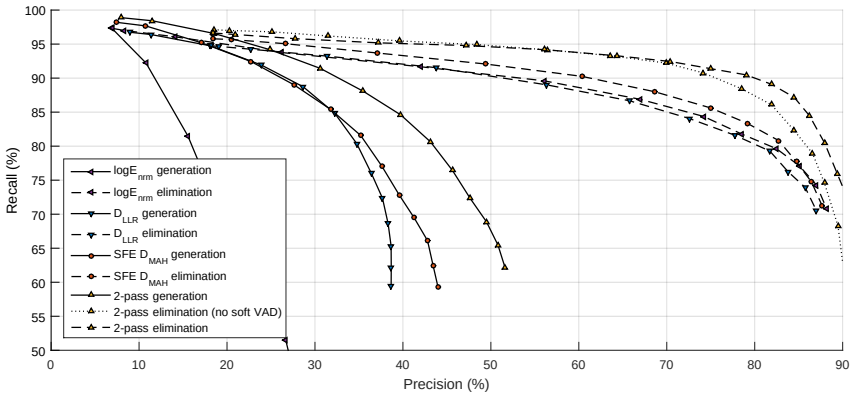

Figure 4: Precision-recall curves of the proposed speaker segmentation methods on the COST278 test data. The error margin is set to 1000ms, and hence focus is on whether real speaker change points are detected or not.

the selection criterion gets more strict. The SFE methods lead to significantly better located change points than the LLR method and $\log E_{\text{nrm}}$ as illustrated in Fig. 3.

The next step is to select optimal operating points on the PR curves of the development data to initialize the boundary elimination. We enforce a minimum average length of the generated speaker segments T_{masl} of 5 seconds for all boundary generation systems except for the $\log E_{\text{nrm}}$ boundary generation. For the latter we enforce $T_{\text{masl}} = 1$s which will result in over-segmentation that should be handled by the subsequent boundary elimination. The parameter λ in Eq. (9) is used to create the precision-recall (PR) curves for the boundary elimination of the one-pass $\log E_{\text{nrm}}$, D_{LLR} and D_{MAH} speaker segmentation systems. The results can again be found in Fig. 3 and Fig. 4.

The over-segmentation of the $\log E_{\text{nrm}}$ boundary generation does not severely deteriorate the subsequent BIC boundary elimination. This causes $\log E_{\text{nrm}}$ to significantly outperform the D_{LLR} method when considering an error margin of 500ms. The more accurate SFE initialization results in significantly better located change points after boundary elimination compared to the $\log E_{\text{nrm}}$ boundary generation.

The final step is to select an operating point on the boundary generation PR curve to initialize the subsequent speaker clustering. The segmentation parameters that result in optimal clustering results and corresponding precision-recall values are presented in Table 1.

6.1.2. Adaptive speaker segmentation

The results of the two-pass system proposed in Section 4.1 are also presented in Fig. 3 and Fig. 4. The system uses SFE boundary generation in both passes and CDS-based boundary elimination in the second pass. The boundary elimination of the second pass is evaluated for different values of the CDS boundary elimination threshold α_{CDS}. We evaluate the system with and without the soft VAD during speaker factor extraction. In the first pass, the system applies PLDA clustering on the output of the BIC boundary elimination. The clustering parameters are tuned to minimize the SER on the development data, see Section 6.2 for more details. Next, we use a threshold $\log E_{\text{nrm}}(t) > -1$

Table 1: Optimal operating points (boundary Precision and Recall) produced by the different boundary elimination modules on the COST278 test data.

segmentation	500ms margin		1000ms margin		boundary elimination
	P(%)	R(%)	P(%)	R(%)	threshold
$\log E_{\text{nrm}}$+BIC	52.1	83.0	56.1	89.6	$\lambda = 4.0$
D_{LLR}+BIC	49.2	77.5	56.3	89.0	$\lambda = 4.0$
D_{MAH}+BIC	57.7	86.1	60.3	90.3	$\lambda = 4.0$
2-pass D_{MAH}+CDS (no VAD)	74.3	83.5	78.5	88.4	$\lambda = 4.0, \alpha_{\text{CDS}} = 0.5$
2-pass D_{MAH}+CDS	78.9	85.5	81.9	89.1	$\lambda = 4.0, \alpha_{\text{CDS}} = 0.5$

to differentiate between speech and non-speech frames and retrain the soft VAD speech and non-speech GMMs accordingly. If soft VAD is disabled we simply retrain the GMM on all frames in the speech regions. Finally, the new eigenvoices are determined on basis of the output clusters.

The introduction of speaker-specific eigenvoices in the two-pass system does not only deliver more accurate estimations of the speaker change points, it also leads to the detection of speaker changes that were initially discarded by the non-adaptive speaker segmentation. More specifically, the two-pass SFE system outperforms the one-pass SFE system for high values of the precision corresponding to longer speaker segments on average. The higher recall in the high precision operating area is mainly caused by the fact that CDS boundary elimination is outperforming the BIC clustering of adjacent speaker segments. This will benefit the global speaker clustering as it will start from these high precision operating points to enforce more data to build the initial speaker models for the AHC. The incorporation of adaptive soft VAD during speaker factor extraction mainly results in the generation of more accurate speaker change points but also in a small improvement of the number of detected change points for high precision operating points.

Table 1 shows that the 2-pass approach allows us to select operating points with comparable recall rates to the one-pass D_{MAH} but with a 20% absolute increase in precision. This results in significantly longer detected speaker turns which should benefit the speaker clustering that follows. The 2-pass system significantly outperforms all speaker segmentation systems reported in (Žibert et al., 2005).

6.2. Speaker clustering

The next section contains a thorough analysis of agglomerative speaker clustering described in Section 3.4, followed by an evaluation of the domain adaptation technique proposed in Section 4.2.

6.2.1. Baseline speaker clustering

In order to evaluate the actual speaker clustering performance of the different systems irrespective of the errors made by the speech/non-speech segmentation or speaker segmentation, we initially feed the different speaker clustering algorithms the sentence boundaries given by ground truth annotations. We do not merge subsequent sentences of the same speaker in the ground truth. This results in a over-segmented initialization

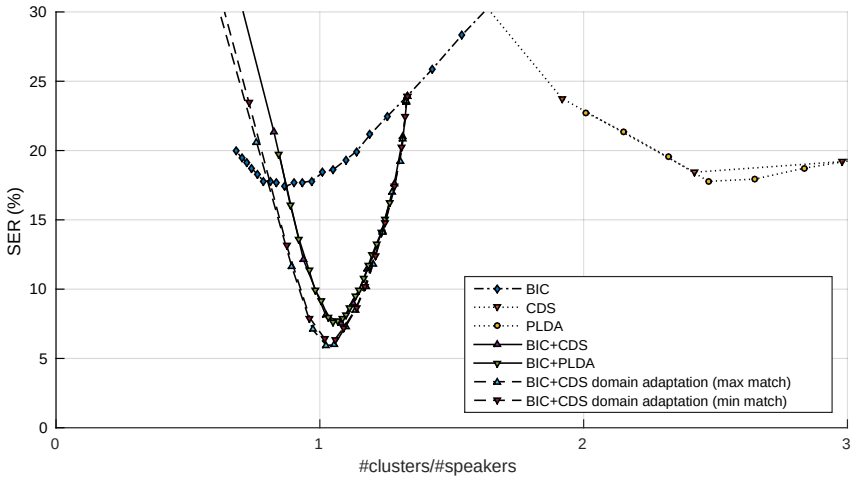

Figure 5: Speaker clustering performance analysis initialized by oracle speaker segments: Speaker Error Rate (SER) vs. cluster to speaker ratio for different cluster algorithms on the COST278 test data.

for the cluster algorithms comparable to the automatic speaker segmentation and thus, in turn, should deliver reliable cluster parameter settings for both scenarios. We compare five AHC systems: (1) BIC clustering (BIC), (2) CDS of speaker factors (CDS), (3) iVector PLDA clustering (PLDA), (4) initial BIC clustering followed by CDS of speaker factors (BIC+CDS) and (5) initial BIC clustering followed by iVector PLDA clustering (BIC+PLDA). We assess the performance by plotting the achieved SER versus the ratio of number of estimated speakers by the number of real speakers by varying the AHC merger threshold. For BIC, CDS and PLDA clustering this corresponds with the parameters λ, β_{CDS} and β_{PLDA} respectively. The results are shown in Fig. 5.

All described clustering techniques operate on the original acoustic feature vectors appended with their Δ-features. The iVector PLDA clustering uses a speech UBM of 256 mixtures and only considers speech frames with $\log E_{\mathrm{nrm}}(t) > 0.5$. The ranks of $\boldsymbol{T}$ and $\boldsymbol{V}$ are set to 100 and 80 respectively. CDS clustering uses the same speech UBM and the rank of the eigenvoice matrix is set to 100.

The one-stage clustering algorithms BIC, CDS, and PLDA are able to achieve a similar optimal SER of about 18%. It is clear however that BIC clustering achieves this level of performance by merging too many clusters and it therefore underestimates the number of speakers. The advanced CDS and PLDA factor analysis techniques on the other hand are not able to process short speaker segments very well. This is expressed by the competitive optimal SER, but gross overestimation of number of speakers at that optimal working point.

Based on this observation, a logical next step is to perform initial BIC clustering with a conservative threshold ($\lambda = 6.0$) and to apply CDS clustering or PLDA clustering in a second stage using the BIC clusters as initialization. The results are again depicted in Fig. 5. This two-stage approach delivers a huge reduction in SER. We see a relative decrease of 58% resulting in a optimal SER of 7.6% for both BIC+CDS and BIC+PLDA

Table 2: Speaker clustering performance (Speaker Error Rate, boundary Precision and Recall in percentage) after initialization by different speaker segmentation modules and initial BIC clustering.

	PLDA clustering					CDS clustering				
error margin		500ms		1000ms			500ms		1000ms	
segmentation	SER	P	R	P	R	SER	P	R	P	R
$\log E_{\text{nrm}}$	10.1	70.5	82.0	75.5	88.2	10.4	70.3	81.8	75.3	88.0
D_{LLR}	10.3	64.4	76.8	73.5	87.9	10.1	64.8	76.8	74.0	87.9
D_{MAH}	10.0	73.9	85.1	77.2	89.1	9.8	74.0	85.0	77.3	89.0
2-pass D_{MAH} (no VAD)	9.6	78.7	82.7	83.1	87.5	9.2	78.8	83.0	83.3	88.0
2-pass D_{MAH}	9.1	81.9	84.9	85.0	88.6	8.8	82.2	84.8	85.3	88.5

clustering. There is only a slight over-estimation of speakers at the optimal working point. The number of clusters by number of speakers ratio is 1.05. The BIC+CDS clustering obtains an optimal SER of 6.0% when it is initialized with oracle segments with subsequent sentences of the same speaker already merged. This gives us an upper limit for the performance we can obtain when the clustering is started with automatically generated speaker segments.

We select the cluster parameters by minimizing the SER on the development data and initialize the clustering with the segmentation outputs. The PLDA cluster threshold β_{PLDA} is set to 2.5, the CDS cluster threshold β_{CDS} is fixed to 0.35. The final results on the test data are listed in Table 2, together with the boundary precision and recall after clustering for the two values of the error margin.

Table 2 shows that an improved segmentation normally results in a reduction of the SER. This is especially true if the segmentation quality is measured with the more strict error margin. BIC+PLDA and BIC+CDS clustering deliver competitive results. The most striking result is that the two-pass system causes a relative drop of the SER by 10% (from 9.8% to 8.8%). This reduced the absolute performance difference with the oracle cluster system with optimal cluster threshold from 3.8% to 2.8%. The integration of soft VAD in the speaker segmentation module plays a significant role in this performance gain. Additional Viterbi resegmentation to refine the boundaries returned mixed results: the LLR-based baseline system showed a small improvement while all other speaker segmentation systems showed small degradations. The speaker error rates of the CDS cluster system per language set of the COST78 data set are shown in Fig. 6. There are no large discrepancies in performance and we achieve a minimum SER of 4.7% and a maximum SER of 12% for the GA and SK language sets respectively. Note that the BE language set is used for parameter tuning which leads to an SER of 4.9%. Using a subset of each language set for development, Silovský and Prazak (2012) achieved an average SER of 11.8% on the BE, CZ, HU, SI, SI2 and SK language sets. Our BIC+CDS clustering achieves an average SER of 8.6% across the same language sets.

6.2.2. Domain adaptation of eigenvoices

To explore the potential of the domain adaptation described in Section 4.2, each language set is divided into two equal parts (according to the number of files) in order to perform two-fold cross-validation. One part is used to train the in-domain eigenvoices

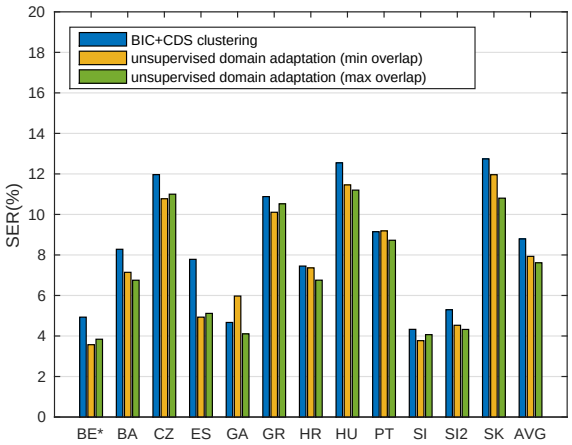

Figure 6: BIC+CDS clustering performance on the COST278 data set per language set initialized with 2-pass speaker segmentation. With and without unsupervised domain adaptation enabled. The * denotes development data.

while the other subset is used for evaluation. Each language set of the COST278 data (partially) consists out of episodes from the same news show and hence several key speakers appear in multiple episodes. We consider two scenarios. In the first scenario, the speaking time of recurring speakers is divided more or less evenly across both sets, hence maximizing the impact of recurring key speakers. In the second scenario we minimize the amount of recurring key speakers by dividing the speaking time of all recurring speakers as unevenly as possible across the two sets. This should give us an upper and lower limit estimation of the performance gains that could be achieved by domain adaptation for speaker clustering. The optimal split results in 10% of the speakers appearing in both sets, which account for 36% of the speaking time. The worst split results in 5% recurring speakers covering 16% of the speaking time. We refer to these two scenarios as *maximum overlap* and *minimum overlap*.

There are two use cases to be considered for domain adaptation of speaker clustering. A *supervised* use case in which speaker labels of the data used for adaptation were manually created or auto-generated diarization output was corrected. And *unsupervised* domain adaptation in which no extra time was invested in correcting this diarization output. In the supervised case we use the ground truth speaker labels of the COST278 data. The unsupervised adaptation uses the speaker labels generated by our top performing BIC+CDS clustering initialized with the 2-pass D_{MAH} speaker segments which achieved an SER of 8.8%. Fig. 5 shows the initial supervised domain adaptation results where the speaker clustering starts from oracle speaker segments. The optimal SER of 7.5% achieved by BIC+CDS clustering drops to 5.9% and 6.4% for the maximum speaker overlap and the minimum speaker overlap respectively. This corresponds with a 21% (15%) relative error reduction in the best (worst) case. The extra speaker-specific eigenvoices clearly help to make a distinction between the speakers inside the domain.

Table 3: Impact of domain adaptation on the BIC+CDS clustering performance initialized by the 2-pass speaker segmentation (Speaker Error Rate in percentage) with the four different test setups.

speaker overlap	maximal		minimal	
no-adaptation	supervised	unsupervised	supervised	unsupervised
8.8	7.6	7.6	7.8	7.9

Also the ratio between the number of detected speakers and the real number of speakers at the optimal SER working point is closer to the ideal value of one. A ratio of 1.02 is achieved with maximum overlap between speakers. The optimal SER of 6.0% achieved by BIC+CDS clustering initialized with oracle segments where subsequent sentences of the same speaker are already merged drops down to 5.0% (5.3%) when domain adaptation is enabled with maximum (minimum) speaker overlap.

The results of the complete work flow with supervised and unsupervised domain adaptation can be found in Table 3. Due to the expanded eigenvoice matrix, the dimension of the speaker factor vectors increased and a new optimal CDS clustering threshold β_{CDS} had to be determined. The threshold is now set to 0.525. In future work more automatic ways of determining the AHC thresholds might be explored, but for now the manual tuning of the parameter on a single development language delivers reasonable results in the broadcast news domain as illustrated by the per language set results shown in Fig. 6.

In case of maximum speaker overlap the SER decreases with 13% relative to 7.6%. As seen in the previous experiments, the domain adaptation only degrades slightly with minimum speaker overlap. There is no significant degradation from using auto-generated speaker labels instead of the oracle speaker labels. The results show that unsupervised domain adaptation is a viable approach to enhance speaker clustering.

6.3. Impact of automatic speech/non-speech segmentation

In this section we revisit the different segmentation methods, but now operating on speech segments generated by the automatic speech/non-speech segmentation of Section 3.2 instead of using the oracle speech/non-speech (SNS) segments. The considered speaker clustering method is BIC+CDS clustering. The different system configurations are evaluated using the Diarization Error Rate (DER). This is the sum of the of a speaker error, false alarm and missed speech component as explained in Section 5.3. However, all these error components are relative to the total duration of the audio making a direct comparison with previously obtained speaker error rates problematic as these were relative to the total ground truth speech duration. Therefore we introduce a compensated SER that considers the percentage of frames that are attributed to a wrong speaker relative to the duration of ground truth speech that was actually classified as speech. This approach provides speaker error rates that can be compared directly. The results can be found in Table 4. The false alarm rate is 0.9% and the missed speech amounts to 1.8%.

We observe identical trends as compared to the systems initialized with oracle SNS segments. The adaptive speaker segmentation with speaker factors (2-pass D_{MAH}) delivers a 13% relative decrease of speaker error rate compared to segmentation based on basic speech activity marks ($\log E_{\text{nrm}}$). Domain adaptation (with maximum speaker overlap between the sets) results in a further relative decrease in SER of 13%. The compensated

Table 4: BIC+CDS clustering performance in percentage after initialization by different speaker segmentation modules. Both oracle or auto-generated speech/non-speech annotations are considered. The compensated SER only considers errors within detected speech that overlaps with ground truth speech. This enables a fair comparison between the SER of the systems initialized by either oracle or auto-generated speech/non-speech annotations.

SNS segmentation	oracle	auto-generated	
speaker segmentation	SER	compensated SER	DER
$\log E_{\text{nrm}}$	10.4	9.8	11.7
D_{LLR}	10.1	9.6	11.5
D_{MAH}	9.8	9.2	11.1
2-pass D_{MAH}	8.8	8.5	10.5
2-pass D_{MAH} + domain adaptation	7.6	7.4	9.4

SER is slightly lower than the SER obtained with oracle speech/non-speech marks. This might be caused by the fact that the SNS segmentation mainly misclassifies degraded speech that is quite hard to assign to one of the speaker clusters in a correct way. The amount of false alarm errors which now have to be assigned to a cluster is not large enough to completely cancel out the positive impact of the more pure speaker clusters. The results also indicate that the automatic speech/non-speech segmentation is robust enough to justify the speaker segmentation on a per detected speech segment basis. The results per language set of BIC+CDS clustering with domain adaptation can be found in Fig. 7. The lowest DER of 5.3% is achieved on the GR language set, the worst performance with a DER of 13.2% is seen on the SI2 language set. The average DER across the complete test set is 9.4%.

It is clear that the introduction of factor analysis techniques for speaker diarization and related adaptation techniques result in a huge performance gain compared to basic BIC-based approaches. The latter achieved DERs in the range of 19% to 35% on the COST278 data (Žibert et al., 2005; Žibert and Mihelič, 2008; Silovský and Prazak, 2012). Note that we achieve a DER of 20% with D_{LLR} speaker segmentation followed by BIC speaker clustering ($\lambda = 10.5$).

7. Conclusions

At the heart of every speaker diarization system is a component that decides whether two speech segments were uttered by the same speaker or by different speakers. To function optimally, this component must be able to get a good idea of the speaker characteristics with only short speech segments available. This in turn requires effective suppression of nuisance factors such as phonetic content, channel and back-ground noise. Total Variability and eigenvoice subspace approaches characterize variable length speech segments with compact fixed length vectors and integrate out most of the phonetic content in the process. This makes them well suited as basic components in a speaker diarization system.

In this paper, we investigated the use of these two subspace methods in the various stages of the speaker diarization process. We compared iVectors –which mainly suppress variability due to phonetic content and hence require additional techniques such as PLDA

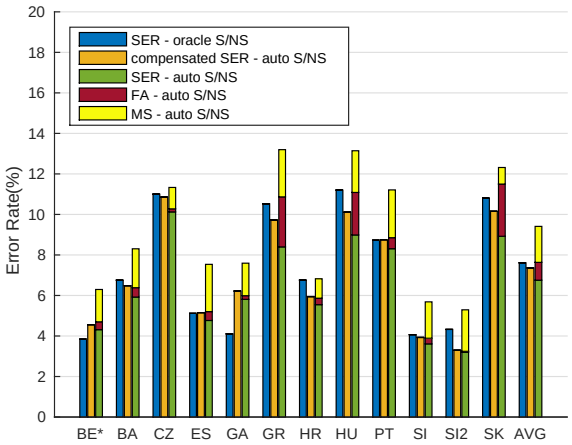

Figure 7: BIC+CDS clustering performance on the COST278 data set per language set with domain adaptation (maximum speaker overlap) initialized by the 2-pass speaker segmentation. Both oracle or auto-generated speech/non-speech annotations are considered. The compensated SER only considers errors within detected speech that overlaps with ground truth speech. This enables a fair comparison between the SER of the systems initialized by either oracle or auto-generated speech/non-speech annotations. The * denotes development data.

to suppress the remaining nuisance factors– with eigenvoices –a subspace method that tries to suppress all non speaker related variability in one step. As with most techniques, some careful design decisions are needed to get the most out of these techniques.

A first set of design decisions involve the trade-off between the precision with which speakers can be characterized (the number of components in the supervector and the number of basis vectors in the subspace method) and the number of frames needed to accurately estimate the weights. This is reflected in our system in the relatively small number of eigenvoices used for speaker change detection in combination with the use of short length overlapping comparison windows (high time resolution of the speaker change detection vs. lower precision of the speaker characterization). In the clustering phase (iVector+PLDA or eigenvoices+CDS), the balancing between precision and amount of data could be relaxed by inserting an initial ΔBIC clustering stage which increases the amount of data available to the subspace method. This approach resulted in a 58% relative reduction in speaker error rate.

Although that robustness to noisy input and suppression of nuisance factors are considered a key-characteristic of the subspace methods, it still proved very helpful to take additional measures such as feature selection and feature normalization. For example, removing (hard VAD) or suppressing (soft VAD) frames that are dominated by the background noise and channel (e.g. inter-word silences or closures in plosives) and hence provide little to no information concerning the speaker, reduced the speaker error rate. In the same vain, further suppression of nuisance factors is achieved by feature warping of the acoustic features to a standard normal distribution. The suppression of nuisance factors in eigenvoices during speaker segmentation was incomplete and the change in pho-

netic content still had an impact on the speaker characterization. However, this source of variability can be modeled on the test file itself and a Mahalanobis-based distance measure can be deployed to emphasize changes induced by other sources, e.g. the targeted speaker change.

Another important aspect proved to be the specificity of the eigenvoices. Most speakers in the training set are only seen in combination with one or a few background conditions. As a result the eigenvoice speaker model is unable to accurately cover all speakers and background conditions encountered in the test data. By employing a two-pass speaker diarization approach, the eigenvoice model can be made to fit the test audio much better. This adaptation process resulted in significant gains for speaker segmentation. It also proved to be a viable approach to domain-adaptation for speaker clustering. This adaptive speaker diarization delivered an speaker error rate of 7.4% on the multilingual COST278 broadcast news data, compared to 9.2% when no adaptation was applied. It is clear that factor analysis (subspace) techniques have become an indispensable part of speaker diarization approaches and pave the way to new straightforward adaptation techniques.

Acknowledgments

This work was supported by IWT Innovatief Aanbesteden within the scope of the VRT STON (Subtitling by using speech and language technology) project.

References

Bansé, D., Doddington, G.R., Garcia-Romero, D., Godfrey, J.J., Greenberg, C.S., Martin, A.F., McCree, A., Przybocki, M.A., Reynolds, D.A., 2014. Summary and initial results of the 2013-2014 speaker recognition i-vector machine learning challenge, in: Proc. Interspeech, pp. 368–372.

Bell, P., Gales, M., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., Woodland, P., 2015. The MGB challenge: Evaluating multi-genre broadcast media recognition, in: Proc. ASRU, pp. 687–693.

Brummer, N., McCree, A., Shum, S., Garcia-Romero, D., Vaquero, C., 2014. Unsupervised domain adaptation for i-vector speaker recognition, in: Odyssey 2014, pp. 260–264.

Burget, L., Matějka, P., Schwarz, P., Glembek, O., Černocký, J., 2007. Analysis of feature extraction and channel compensation in GMM speaker recognition system. IEEE Trans. Audio, Speech and Language Processing 15, 1979–1986.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., Wellner, P., 2006. The AMI meeting corpus: A pre-announcement, in: Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction, pp. 28–39.

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

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

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech and Language Processing 19, 788–798.

Desplanques, B., Demuynck, K., Martens, J.P., 2014. Robust language recognition via adaptive language factor extraction, in: Proc. Interspeech, pp. 2160–2164.

Desplanques, B., Demuynck, K., Martens, J.P., 2015. Factor analysis for speaker segmentation and improved speaker diarization, in: Proc. Interspeech, pp. 3081–3085.

Desplanques, B., Demuynck, K., Martens, J.P., 2016. Soft vad in factor analysis based speaker segmentation of broadcast news, in: Odyssey 2016: The Speaker and Language Recognition Workshop, pp. 158–165.

Desplanques, B., Martens, J.P., 2013. Model-based speech/non-speech segmentation of a heterogeneous multilingual TV broadcast collection, in: Proc. ISPACS, pp. 55–60.

Ferrer, L., McLaren, M., Lawson, A., Martin, G., 2015. Mitigating the effects of non-stationary unseen noises on language recognition performance, in: Proc. Interspeech, pp. 3446–3450.

Ferrer, L., McLaren, M., Scheffer, N., Lei, Y., Graciarena, M., Mitra, V., 2013. A noise-robust system for NIST 2012 speaker recognition evaluation., in: Proc. Interspeech, pp. 1981–1985.

Fox, E., Sudderth, E., Jordan, M., Willsky, A., 2011. A sticky HDP-HMM with application to speaker diarization. Annals of Applied Statistics 5, 1020–1056.

Garcia-Romero, D., Espy-Wilson, C.Y., 2011. Analysis of i-vector length normalization in speaker recognition systems, in: Proc. Interspeech, pp. 249–252.

Garcia-Romero, D., McCree, A., 2014. Supervised domain adaptation for i-vector based speaker recognition, in: Proc. ICASSP, pp. 4047–4051.

Garofolo, J., Fiscus, J., Fisher, W., 1997. Design and preparation of the 1996 hub-4 broadcast news benchmark test corpora, in: Proc. of DARPA Speech Recognition Workshop, pp. 15–21.

Glembek, O., Burget, L., Matějka, P., Karafiát, M., Kenny, P., 2011. Simplification and optimization of i-vector extraction, in: Proc. ICASSP, pp. 4516–4519.

Karanasou, P., Gales, M.J.F., Lanchantin, P., Liu, X., Qian, Y., Wang, L., Woodland, P.C., Zhang, C., 2015. Speaker diarisation and longitudinal linking in multi-genre broadcast data, in: Proc. ASRU, pp. 660–666.

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

Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2007. Joint factor analysis versus eigenchannels in speaker recognition. IEEE Transactions on Audio, Speech and Language Processing 15, 1435–1447.

Kenny, P., Stafylakis, T., Ouellet, P., Gupta, V., Alam, J., 2014. Deep neural networks for extracting baum-welch statistics for speaker recognition, in: Proceedings of Odyssey 2014: The Speaker and Language Recognition Workshop, pp. 293–298.

Le Lan, G., Charlet, D., Larcher, A., Meignier, S., 2016. Iterative PLDA adaptation for speaker diarization, in: Proc. Interspeech, pp. 2175–2179.

McLaren, M., Graciarena, M., Lei, Y., 2015. Softsad: Integrated frame-based speech confidence for speaker recognition, in: Proc. ICASSP, pp. 4694–4698.

Ning, H., Liu, M., Tang, H., Huang, T., 2006. A spectral clustering approach to speaker diarization, in: Proc. ICSLP.

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

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

Reynolds, D.A., 1997. Comparison of background normalization methods for text-independent speaker verification, in: Proc. Eurospeech, pp. 963–966.

Sell, G., Garcia-Romero, D., 2014. Speaker diarization with PLDA i-vector scoring and unsupervised calibration, in: Proc. SLT, pp. 413–417.

Sell, G., Garcia-Romero, D., McCree, A., 2015. Speaker diarization with i-vectors from DNN senone posteriors, in: Proc. Interspeech, pp. 3096–3099.

Sell, G., McCree, A., Garcia-Romero, D., 2016. Priors for speaker counting and diarization with ahc, in: Proc. Interspeech, pp. 2194–2198.

Senoussaoui, M., Kenny, P., Stafylakis, T., Dumouchel, P., 2014. A study of the cosine distance-based mean shift for telephone speech diarization. IEEE Trans. Audio, Speech and Language Processing 22, 217–227.

Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D.A., Glass, J.R., 2011. Exploiting intra-conversation variability for speaker diarization, in: Proc. Interspeech, pp. 945–948.

Shum, S., Dehak, N., Glass, J., 2012. On the use of spectral and iterative methods for speaker diarization, in: Proc. Interspeech, pp. 482–485.

Silovský, J., Prazak, J., 2012. Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring, in: Proc. ICASSP, pp. 4193–4196.

Silovský, J., Prazak, J., Cerva, P., Zdánský, J., Nouza, J., 2011. PLDA-based clustering for speaker diarization of broadcast streams, in: Proc. Interspeech, pp. 2909–2912.

Stafylakis, T., Katsouros, V., Carayannis, G., 2010. The segmental Bayesian Information Criterion and its applications to speaker diarization. Selected Topics in Signal Processing, IEEE Journal of 4, 857 – 866.

Tang, H., Chu, S.M., Huang, T.S., 2009. Generative model-based speaker clustering via mixture of von

mises-fisher distributions, in: Proc. ICASSP, pp. 4101–4104.
Vandecatseye, A., Martens, J.P., 2003. A fast, accurate and stream-based speaker segmentation and clustering algorithm, in: Proc. Eurospeech, pp. 941–944.
Vandecatseye, A., Martens, J.P., Neto, J., Meinedo, H., Garcia-Mateo, C., Dieguez, J., Mihelic, F., Zibert, J., Nouza, J., David, P., Pleva, M., Cizmar, A., Papageorgiou, H., Alexandris, C., 2004. The COST278 pan-European broadcast news database, in: Proc. LREC, pp. 873–876.
Verwimp, L., Desplanques, B., Demuynck, K., Pelemans, J., Lycke, M., Wambacq, P., 2016. STON: Efficient subtitling in dutch using state-of-the-art tools, in: Proc. Interspeech, pp. 780–781.
Žibert, J., Mihelic, F., Martens, J.P., Meinedo, H., Neto, J.a.o., Docio, L., Garcí a Mateo, C., David, P., Zdansky, J., Pleva, M., Cizmar, A., Zgank, A., Kacic, Z., Teleki, C., Vicsi, K., 2005. The COST278 broadcast news segmentation and speaker clustering evaluation: overview, methodology, systems, results, in: Proc. Interspeech, pp. 629–632.
Žibert, J., Mihelič, F., 2008. Novel approaches to speaker clustering for speaker diarization in audio broadcast news data, in: Speech Recognition. doi:10.5772/6386.
Zelenák, M., Schulz, H., Hernando, J., 2012. Speaker diarization of broadcast news in albayzin 2010 evaluation campaign. EURASIP Journal on Audio, Speech, and Music Processing 2012, 19.
Zhu, X., Barras, C., Meignier, S., Gauvain, J.L., 2005. Combining speaker identification and BIC for speaker diarization, in: Proc. Interspeech, pp. 2441–2444.