Soft VAD in Factor Analysis Based Speaker Segmentation of Broadcast News

Brecht Desplanques, Kris Demuynck, Jean-Pierre Martens

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

Abstract

In this work we propose to integrate a soft voice activity detection (VAD) module in an iVector-based speaker segmentation system. As speaker change detection should be based on speaker information only, we want it to disregard the nonspeech frames by applying speech posteriors during the estimation of the Baum-Welch statistics. The speaker segmentation relies on speaker factors which are extracted on a frame-by-frame basis using an eigenvoice matrix. Speaker boundaries are inserted at positions where the distance between the speaker factors at both sides is large. A Mahalanobis distance seems capable of suppressing the effects of differences in the phonetic content at both sides, and therefore, to generate more accurate speaker boundaries. This iVector-based segmentation significantly outperforms Bayesian Information Criterion (BIC) segmentation methods and can be made adaptive on a file-by-file basis in a two-pass approach. Experiments on the COST278 multilingual broadcast news database show significant reductions of the boundary detection error rate by integrating the soft VAD. Furthermore, the more accurate boundaries induce a slight improvement of the iVector Probabilistic Linear Discriminant Analysis system that is employed for speaker clustering.

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). The number of applications that can benefit from this extra information is numerous, but in this work the focus is on the semi-automatic creation of subtitles. The public broadcaster of Flanders VRT wants to speed up this subtitling process by employing speech technology. As the subtitles must be of a very high quality and as the spoken language has to be converted to a compact written form, 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.

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 representing single speakers. The latter is necessary to add informative color codes to the generated subtitles and to profit from speaker adapted models during speech recognition. In this paper we focus on improving the 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. Although such improvements 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.

In previous work [1] we replaced a speaker segmenter based on the computation of log-likelihood ratios (LLR) and Bayesian Information Criterion (BIC) distances [2] in the acoustic feature space by a segmenter that operates in the so-called speaker factor space. The boundary detection could be enhanced because the phonetic variability that can partially mask the speaker variability can be better suppressed in the speaker factor space. The speaker factor extraction (SFE) method, which is reviewed in Section 2.1.2, follows a paradigm that very much resembles the iVector paradigm [3].

In what follows we revisit the conventional LLR-BIC-based method and modify it so that its distances can follow from the statistics of overlapping windows. This concept of overlapping windows was found to improve the SFE system we developed and we argue here why that is, and why it should also help to improve the LLR-BIC method. The experimental study demonstrates that overlapping windows do lead to a significant improvement but that this improvement is insufficient to close the performance gap between the LLR-BIC system and the SFE system. The latter observation strengthens us in our conviction that working in the speaker factor space offers an additional advantage. A more detailed analysis of the experimental results also revealed that overlapping windows do not so much increase the number of detected speaker boundaries, but rather they put the detected boundaries closer to the positions where human annotators would mark them.

The main contribution of our paper is that it proposes to further improve the SFE method by differentiating between true speech frames and frames which belong to short silences between words or syllables that are always present in the speech segments that must be analyzed. The proposal is to include a soft voice activity detection (VAD) pre-processing step that generates frame-wise speech probabilities, and to employ these probabilities in order to suppress the impact of the ‘nonspeech-like’ frames on the speaker factors. The latter is achieved by weighting each frame with its speech posterior during the calculation of the Baum-Welch statistics (needed for the speaker factor extraction). 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. [4, 5]).

During our experimental study we conduct experiments on the COST278 multilingual broadcast news data set [6] in order to assess the impact of using overlapping windows. We also study the effect of soft VAD on the accuracy of the generated speaker boundaries. Both the boundary accuracy before and after clustering are considered and also the impact of the improved boundary generation on the speaker clustering accuracy is investigated.

2. System architecture

We assume that the speaker diarization system is preceded by a speech/non-speech module which divides the audio into long non-speech segments (having a length of at least 1.5 seconds) interleaved with speech segments. 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 diarization system works on 10ms frames and per frame it gets 16 MFCCs and a normalized log-energy. The latter is defined as

$$\log E_{\text{nrm}}(t) = \log E(t) - \overline{\log E(t)} \tag{1}$$

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 with a time constant of 5 seconds. See [7] for more details.

2.1. Speaker segmentation

The segmentation into homogeneous speaker segments is achieved by means of a two-stage procedure, as explained in [1] and [8]. The first stage generates boundaries on the basis of a sliding window approach, whereas 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.

2.1.1. LLR boundary generation

Candidate change points are generated at places of maximum difference between the statistical distributions of the acoustic vectors in two fixed-length windows (N_w frames) to the left and to the right. In this particular case, the popular ΔBIC criterion reduces to a log-likelihood ratio:

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

with each $\mathbf{\Sigma}$ representing the Maximum Likelihood (ML) estimate of the full covariance matrix of the acoustic features in the corresponding window.

To avoid the detection of spurious peaks, LLR(t) (as a function of 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 $N_{\text{p}}(\mathcal{S})$ 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 $\mathcal{S}$:

$$N_{\text{p}}(\mathcal{S}) = \max(N_{\text{p,min}}, \left\lceil \frac{T(\mathcal{S})}{T_{\text{masl}}} \right\rceil) \tag{3}$$

$N_{\text{p,min}}$ 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). We also prevent the system from generating speaker segments that are shorter than T_{min} (fixed to 1 second).

2.1.2. Boundary generation with speaker factor extraction

In [1] we proposed a more advanced method for speaker segmentation. It performs an online speaker factor extraction (SFE) based on eigenvoices [9] to produce speaker factors for each frame.

If $\boldsymbol{x}_t$ is the extracted speaker factor and if $\boldsymbol{m}$ is the supervector of an UBM of the acoustic feature vectors encountered in the training data, then the GMM supervector $\boldsymbol{m}_t$ representing a window centered around the considered frame is approximated by

$$\boldsymbol{m}_t = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{x}_t \tag{4}$$

with $\boldsymbol{V}$ being the eigenvoice matrix. The speaker factor extraction at time t considers the frames inside a window of length T_{e} centered around t. The procedure for extracting the speaker factors is similar to the iVector extraction described in [10].

The matrix $\boldsymbol{V}$ contains R eigenvoices obtained on the training data. The eigenvoices are obtained by means of Principal Component Analysis (PCA) initialization [11] followed by a number of iterations of the non-simplified Expectation-Maximization algorithm described in [10]. We do not use the Total Variability framework as 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 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. In order to constrain the computational efficiency we use an UBM with a low number of mixtures (=32) and a low rank 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

$$\Delta\boldsymbol{x}_t = \boldsymbol{x}_{t-\tau} - \boldsymbol{x}_{t+\tau} \tag{5}$$

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 ($T_{\text{e}} = 1\text{s}$), 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 $T_{\mathbf{\Sigma}}$ 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 $\mathbf{\Sigma}_{L,t}$. 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 $\mathbf{\Sigma}_{R,t}$. Under these hypotheses the covariance matrices $\mathbf{\Sigma}_L$ and $\mathbf{\Sigma}_R$ 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)

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

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 $\mathbf{\Sigma}_{L(R)}$ is measured on the test data itself, the approach is presumed to be insensitive to mismatches between training and test data.

The search for peaks in $D_{\text{MAH}}(t)$ is done with the same algorithm that was used for searching the peaks in $D_{\text{LLR}}(t)$.

2.1.3. 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 BIC criterion, 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 BIC distance between two segments is given by

$$\Delta BIC = (N_L + N_R)\log|\mathbf{\Sigma}_{L+R}| \\ - N_L \log|\mathbf{\Sigma}_L| - N_R \log|\mathbf{\Sigma}_R| - \lambda P \quad (7)$$

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

$$P = \frac{1}{2}\left(d + \frac{1}{2}d(d+1)\right)\log\left(N_L + N_R\right) \quad (8)$$

where d is the dimension of the feature vectors. 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 positive. Obviously, at every merge, the ΔBIC values of the endpoints of the newly formed segment have to be updated. The parameter λ in (7) controls the number of boundaries that will be eliminated.

2.2. Agglomerative speaker clustering

The detected speaker segments are finally clustered using the two-stage Agglomerative Hierarchical Clustering (AHC) approach proposed in [12]. Each cluster is supposed to encompass all the segments of a particular speaker.

2.2.1. Initial BIC clustering

In the first stage, some clusters may still be small (few data points), and hence robust techniques such as BIC are preferred. 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 turns out to be negative.

2.2.2. Final PLDA clustering

In the second stage, more advanced techniques are used. First of all, it discards the frames with a low $\log E_{\text{nrm}}$ because these frames can be dominated by background noise. Second, the acoustic features of the selected frames are normalized by means of Feature Warping [13]. Third, iVector PLDA is used to iteratively merge the BIC clusters on the basis of these normalized feature vectors.

The main idea behind the proposed iVector PLDA clustering is that there are different sources of variability between clusters (speaker, channel, phonetic content,...) and that the emphasis should be on the variability that is induced by changes of the speaker. We use Total Variability (TV) [3] to model as much of the 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

$$\boldsymbol{m}_c = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{x}_c \quad (9)$$

where $\boldsymbol{m}$ is the supervector of the Universal Background Model (UBM) of speech and $\boldsymbol{x_c}$ is the fixed length iVector that contains all relevant information concerning cluster c. The procedure for extracting iVectors is described in [10]. 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 in a similar fashion as described in Section 2.1.2. During the training we do not pool speaker segments belonging to the same speaker however, and all segments are kept separate.

Once the iVectors are extracted, it is time to highlight their speaker-specific component. If the dimensionality of the vectors is sufficiently small the latter can be achieved by adopting a modified PLDA framework [14]. After whitening and length normalization [15] each iVector is modeled as

$$\boldsymbol{x}_c = \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{y}_c + \boldsymbol{\epsilon}_r \quad (10)$$

where $\boldsymbol{\mu}$ is a global offset, where $\boldsymbol{V}$ represents the basis of the speaker-specific subspace and where $\boldsymbol{y}_c$ 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 $\mathbf{\Sigma}_r$.

The AHC clustering is now controlled by the log-likelihood ratio

$$\text{LLR}_{\text{PLDA}}(c_i, c_j) = \log\frac{p(\boldsymbol{x}_{c_i}, \boldsymbol{x}_{c_j}|\mathcal{H}_s)}{p(\boldsymbol{x}_{c_i}|\mathcal{H}_d)p(\boldsymbol{x}_{c_j}|\mathcal{H}_d)} \quad (11)$$

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. When the two most similar clusters are being merged a new iVector has to be computed for the new cluster. The clustering process is terminated when all log-likelihood ratios fall below a predetermined threshold β.

2.3. Two-pass system

The eigenvoice model of Section 2.1.2 is determined on training data which may not really match the evaluation data. In combination with the fact that we use low-dimensional models (for computational reasons), degraded speaker segmentation models may emerge. This model mismatch can be eliminated by a cascade of two segmentation and clustering systems each embedding the same SFE segmentation algorithm but a different UBM and different eigenvoices. There are also arguments for choosing another boundary elimination criterion in the two segmenters.

2.3.1. Update the eigenvoices

First we adapt the UBM to model the speech frames of the analyzed file better. Then we use the speaker clusters emerging from the first pass 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 perfectly match the speakers in the file, the speaker factors may also be much 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.

2.3.2. CDS boundary elimination

In [1] we showed that Cosine Distance Scoring (CDS) outperforms BIC in the boundary elimination stage of a speaker segmenter whenever the eigenvoices match the test data. Consequently, we propose to use CDS as the criterion for eliminating speaker boundaries in the final segmenter. 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:

$$CDS(s_L, s_R) = 1 - \frac{\boldsymbol{x}_{s_L} \cdot \boldsymbol{x}_{s_R}}{\|\boldsymbol{x}_{s_L}\|\|\boldsymbol{x}_{s_R}\|} \quad (12)$$

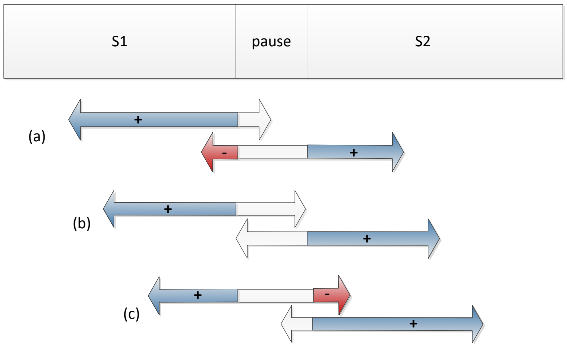

Figure 1: *How the use of overlapping windows tend to produce one peak in the dissimilarity measure in the middle of a pause between speaker turns. Blue regions (+) contribute to the dissimilarity, whereas red regions (−) reduce dissimilarity.*

The elimination continues until the lowest CDS value falls under a predefined threshold α.

3. Proposed modifications

3.1. LLR boundary generation with overlapping windows

In experiments, we established that the LLR boundary generation was not working properly if $D_{\text{LRR}}(t)$ was not smoothed by a moving average filter. The SFE method on the other hand needed much less smoothing of $D_{\text{MAH}}(t)$. We argue that the main reason for this discrepancy is the fact that the SFE method compares information in overlapping windows whereas the LLR method compares adjacent windows.

In fact, a significant fraction of the speaker changes is characterized by a short inter-speaker pause. The acoustic content in adjacent windows tend to be the most dissimilar at positions coinciding with the beginning or the end of the pause. Thus, the pause is likely to induce two major peaks in the raw $D_{\text{LLR}}(t)$, therefore introducing a false positive. By smoothing $D_{\text{LLR}}(t)$ with a moving average filter of sufficient length, the two former maximums will disappear and make place for one maximum in the middle of the pause.

By working with overlapping windows, one can induce a very similar effect. Consider Figure 1 which shows two speaker segments separated by a pause and which shows the overlapping windows considered by SFE boundary generation for three positions: (a) at the beginning, (b) in the middle and (c) at the end of the pause respectively. The blue regions (+) indicate the frames responsible for increasing dissimilarity between the comparison windows. The red regions (−) indicate areas that decrease the speaker dissimilarity as they imply frames of the two speakers being present in one window. The net contribution to the dissimilarity is maximal when t lies in the middle of the pause. Consequently, the distance measure is maximal in the center of the pause. Obviously, if the overlap becomes too large both compared windows will always contain information of the two speakers, making the speaker boundary more difficult to detect. It is easy to verify that in the absence of a pause between speakers the overlapping boundary generation will continue to produce one peak at the position of the speaker change.

In the experimental section we will investigate whether LLR in combination with overlapping comparison windows can compete with SFE or whether SFE is fundamentally better than LLR. Note that for the estimation of Σ_{L+R} in Equation (2) the frames in the overlap region will be considered only once.

3.2. Soft VAD for speaker segmentation

It remains a weakness of the SFE-based segmentation algorithm that it does not differentiate between speech frames and non-speech frames, because the non-speech frames are not expected to contribute information concerning the speaker identity. We therefore propose to include a soft voice activity detector (VAD) that generates frame-wise speech posteriors. Instead of treating all speech frames equally, frames with a high speech posterior will get more weight during the speaker factor extraction.

We prefer a soft VAD over a hard VAD which makes binary decisions because the former ensures that a speech frame that is getting a low speech posterior can still contribute to the speaker factor extraction. However, an even stronger argument in disfavor of a hard VAD is that such a VAD would generate the same speaker factor values for a number of consecutive frames, and this may pose a serious challenge for the estimation of the covariance matrices needed for computing D_{MAH}.

The soft VAD is integrated in the SFE boundary generation and in the CDS boundary elimination stage embedded in the second pass of the 2-pass system. The soft VAD was also integrated in the iVector-based speaker clustering, but this did not result in any performance gains with respect to the baseline process of selecting only high-energy speech frames.

3.2.1. GMM-based soft VAD

We chose to implement the soft VAD using a simple GMM-based approach [16]. This involves the training of a speech GMM θ_S and a non-speech GMM θ_{NS} on some training data. The speech GMM is trained on the high-energy frames (high $\log E_{\text{nrm}}$) found in the speech segments. The non-speech training data is created by pooling the low-energy speech frames and the non-speech frames.

During evaluation we extract a speech posterior $p(\theta_S|\boldsymbol{o}_t)$ for frame vector $\boldsymbol{o}_t$ by transforming the speech and non-speech log-likelihoods $\log p(\boldsymbol{o}_t|\theta_{S/NS})$ as follows:

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

However, in all experiments we will assume equal priors for both classes. 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 $\rho = 1$.

The soft VAD is integrated into the SFE by slightly modifying the estimation of the zero- and first-order Baum-Welch statistics to

$$N^m_{\mathcal{X}} = \sum_{\boldsymbol{o}_t \in \mathcal{X}} p(\theta_S|\boldsymbol{o}_t) \, \gamma(\theta_{S,m}|\boldsymbol{o}_t) \quad (14)$$

$$\boldsymbol{f}^m_{\mathcal{X}} = \sum_{\boldsymbol{o}_t \in \mathcal{X}} p(\theta_S|\boldsymbol{o}_t) \, \gamma(\theta_{S,m}|\boldsymbol{o}_t) \, \boldsymbol{o}_t \quad (15)$$

Each frame is weighted by its speech posterior and $\gamma(\theta_{S,m}|\boldsymbol{o}_t)$ is the occupation probability of mixture m of the speech GMM. $\mathcal{X}$ is the relevant set of frames for which the speaker factors are extracted. The modified Baum-Welch statistics are used during the frame-wise SFE as well as during the CDS boundary elimination.

3.2.2. Two-pass system model retraining

To enable soft VAD in the two-pass system, we need to retrain both the speech GMM and the non-speech GMM. 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 on the other hand are extracted across all frames of the speech regions belonging to the considered speaker cluster.

An alternative for the energy-based speech/non-speech frame selection would have been to select the speech frames using the soft VAD detector that was embedded in the first pass of the segmentation system.

4. Experiments and results

In all experiments the speaker segmentation and speaker clustering start from an oracle speech/non-speech (SNS) annotation. The annotation protocol specified that only non-speech segments with a duration of more than 1.5 second had to be marked as non-speech, which should also be achievable by automatic SNS segmentation (e.g. [17]). The speaker segmentation is performed independently for each annotated speech segment. The cluster stage works file per file and considers all segments emerging from the speaker segmentation.

4.1. Data

4.1.1. Training data

All models are trained on 66 hours of speech from the 1996 HUB4 Broadcast News training data (3009 speakers).

4.1.2. Development and evaluation data

The evaluation corpus is the multilingual COST278 corpus[1]. It is composed of complete TV news shows broadcasted by 16 European TV stations. It covers 9 national and 2 regional languages. The corpus is divided into 12 language sets (but there are two Slovenian sets) of about three hours each. The BE language set was set aside for parameter tuning and the 11 remaining language sets were used for evaluation. The evaluation data consists of 4386 speaker segments. Please consult [6] and the website for more details about the corpus.

4.2. Evaluation measures

For the evaluation of the speaker segmentation the computed and the annotated speaker change points are linked to oneanother if the gap between them is not larger than a given margin. Unless stated otherwise, we consider an error margin of 500ms. The formed links constitute the basis for deriving the *recall* (= percentage of annotated boundaries that were mapped to a computed boundary) and the *precision* (= percentage of computed boundaries that were mapped to a real boundary). The initial and final points of the annotated speech fragments (speech/non-speech boundaries) are excluded from the evaluation because they would always turn out to be correct in our experiment.

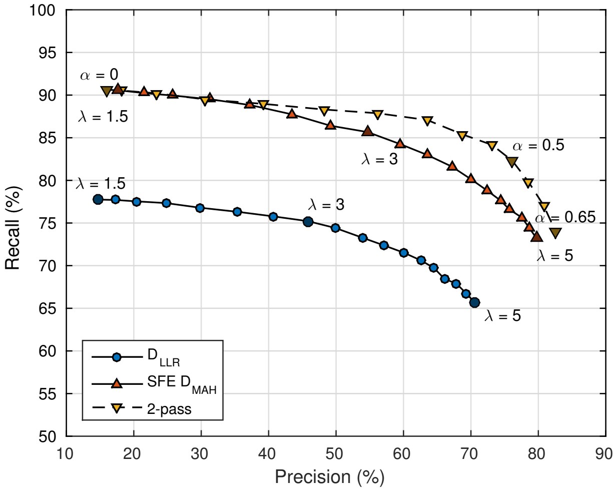

Figure 2: *Precision-recall curves for all baseline speaker segmentation systems.*

For evaluating the diarization system as a whole, we consider the Speaker Error Rate (SER), defined as the percentage of frames that was attributed to a wrong speaker given an optimal mapping (see [18]) between the speaker clusters and the reference annotation.

4.3. Baseline speaker segmentation systems

In this section we assess the speaker segmentation performances of our three baseline systems: (1) LLR boundary generation + BIC-based boundary elimination, (2) SFE boundary generation + BIC-based boundary elimination and (3) a two-pass system working with SFE boundary generation in both passes.

The LLR boundary generation parameters are: the window size N_w (fixed to 200 frames or 2s), the moving average window length N_{avg} (fixed to 300 frames), the minimum number of peaks per speech segment $N_{\text{p,min}}$ (fixed to 3), the enforced minimum average length of the generated speaker segments T_{masl} (fixed to 5 seconds) and a minimum segment duration T_{min} (fixed to 1 second).

The SFE boundary generation parameters are: the number of mixtures of the UBM (fixed to 32), the rank R of $\boldsymbol{V}$ (fixed to 20), the speaker factor extraction window size T_{e} (fixed to 1s), the time difference τ (fixed to 250 ms), the window size $T_{\boldsymbol{\Sigma}}$ used for estimating $\boldsymbol{\Sigma}_L$ and $\boldsymbol{\Sigma}_R$ in Equation (6) (1750ms) and the moving average window length N_{avg} (fixed to 150 frames). The parameter settings for the clustering in the two-pass system can be found in Section 4.6.

The parameter λ in Equation (7) is used to create the precision-recall (PR) curves of the systems. The two-pass system is evaluated for different values of the CDS boundary elimination threshold α instead. All segmentation parameters are tuned to get an optimal PR curve on the development data. The PR curves of the test data are depicted in Figure 2.

In accordance with [1], the SFE-based method significantly outperforms the LLR-based method. Furthermore, 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 latter is relevant as the speaker clustering will start from operating points with a high precision. The maximum recalls for the three systems are 76.6%, 90.2% and 90.7% respectively, which shows that a small fraction of the speaker changes are omitted by the boundary generation. Note that the

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

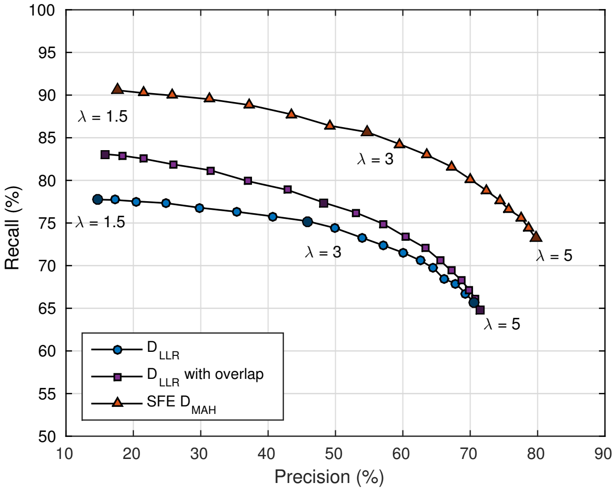

Figure 3: *Precision-recall curves for the modified LLR boundary generation generation. The error margin is set to 500ms.*

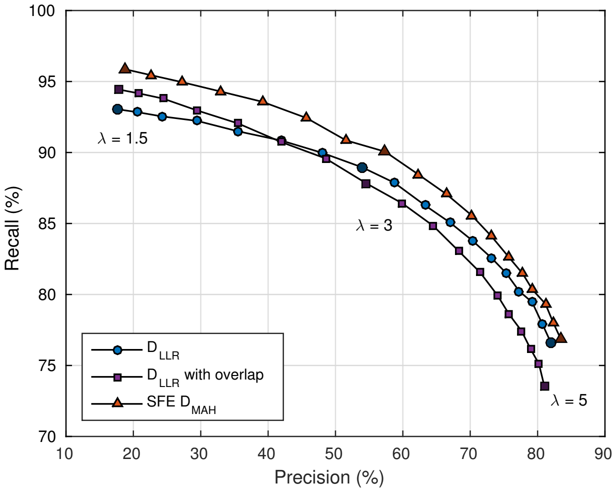

Figure 4: *Precision-recall curves for the modified LLR boundary generation generation. The error margin is set to 1000ms.*

LLR-based system performance is higher than reported in [1]. This follows from the fact that we have now used a hamming averaging window with an optimal length in the boundary generation stage.

4.4. LLR boundary generation with overlapping windows

For the LLR boundary generation with overlapping windows we set the overlap to 50 frames (500ms) and we reduced N_{avg} to 75 frames. PR curves are generated for error margins of 500ms (Figure 3) and 1000ms (Figure 4). The more strict 500ms margin curves reveal how accurate the positions of the boundaries are as the calculated and real boundaries are required to be closely positioned to each other. Whereas the broad 1000ms margin curves reveal how many speaker changes are actually detected.

First of all, the SFE method leads to significantly better located change points than the best LLR method, but it does not detect that many more change points. Likewise, the utilization of overlapping windows in the LLR method improves the positions of the change points without really increasing the number of detected change points. The data show that only a small part of the performance gap between the LLR and the SFE method can be closed by introducing overlapping windows in the LLR segment generation. This proves that the SFE method is more powerful in extracting speaker-specific information from the acoustics.

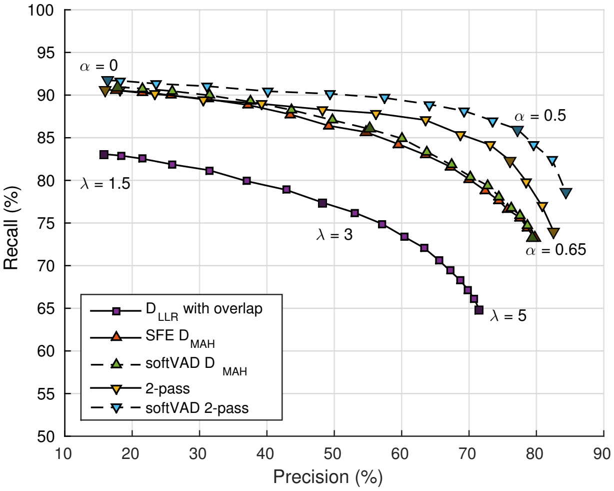

Figure 5: *Precision-recall curves of the soft VAD systems compared to the other segmentation methods. The error margin is set to 500ms.*

4.5. Soft VAD for SFE speaker segmentation

In this section we study the impact of integrating soft VAD in the speaker factor extraction. We maintain the parameter settings of the baseline SFE segmentation. Again we generate the PR-curves for error margins of 500ms and 1000ms. They are depicted in Figures 5 and 6 respectively.

A bit unexpectedly, the soft VAD SFE segmentation does not yield any improvement over the baseline. One possible explanation may be that due to the VAD the speaker factors can be based on a relatively low number of prominent speech frames. In that case it may be better to consider all speech frames, including those that are seriously affected by the background already.

When the soft VAD is integrated in the 2-pass system it does induce a significant gain over the baseline 2-pass system. The VAD does not only lead to more accurate boundaries (500ms margin), but it also leads to more speaker changes being detected (1000ms margin). In line with the hypothesis forwarded above, the performance gap with the baseline 2-pass system is larger for operating points with a high precision (= longer speaker segments), indicating that the CDS boundary elimination especially benefits from the VAD if the speaker segments are long enough.

4.6. Speaker clustering

In this section we initialize the clustering with the speaker segments emerging from the different segmentation systems we tested and we evaluate the Speaker Error Rate (SER).

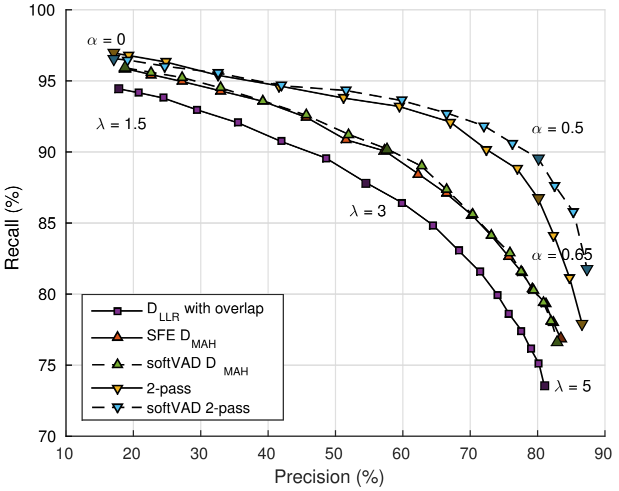

Figure 6: *Precision-recall curves of the soft VAD systems compared to the other segmentation methods. The error margin is set to 1000ms.*

error margin		500ms		1000ms	
	SER(%)	P(%)	R(%)	P(%)	R(%)
D_{LLR}	10.4	64.9	74.1	76.2	87.3
D_{LLR} with overlap	10.1	65.2	75.8	73.8	85.9
D_{MAH}	9.7	74.3	84.2	77.8	88.4
softVAD D_{MAH}	9.8	74.7	84.8	77.9	88.9
2-pass	9.8	79.7	81.3	83.7	85.7
softVAD 2-pass	8.9	81.7	85.0	84.9	88.6

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

The iVector PLDA clustering uses an speech UBM of 256 mixtures and the ranks of $\boldsymbol{T}$ and $\boldsymbol{V}$ is set to 100 and 80 respectively. We also include extra information by adding the Δ-features tot the acoustic feature vector. The threshold λ of the initial BIC clustering is set to 4.5 and the PLDA cluster threshold β is set to 2.5. The clustering starts from the segmentation results obtained with $\lambda = 3.0$ or $\alpha = 0.5$. These cluster parameters are obtained by minimizing the SER on the development data. The final results on the test data are listed in Table 1, together with the boundary precision and recall after clustering for two values of the error margin.

The Table shows that an improved segmentation normally results in an improvement of the SER as well. This is especially true if the segmentation quality is measured with the shorter error margin. The most striking result is that the soft VAD in the two-pass system causes a drop of the SER by nearly 10% (from 9.8% to 8.9%). This drop is actually more important than the raise in precision and recall of the speaker segmentation it provides.

We also tried to further improve our results by adding a Viterbi resegmentation step, but this was not a big success. We did obtain a small improvement for our LLR-based baseline system, but for our SFE-systems we obtained a small degradation instead.

5. Conclusions

We showed that speaker segmentation methods incorporating factor analysis are capable of highlighting speaker-specific information that is hard to discover using conventional methods working directly in the acoustic feature space. Our speaker factor based system produces much more accurate speaker boundaries (deviation from annotations less than 500ms) than our LLR-BIC baseline. Furthermore, by suppressing the importance of non-speech frames in the speaker factor extraction it is possible to get an additional performance gain, at least if it is used in combination with models that were adapted to the file under analysis on the basis of the results obtained with a non-adapted system. Finally, we provided evidence that a more accurate speaker segmentation on its turn also offers a more accurate speaker clustering. For our best system, the speaker error rate could be reduced by approximately 10% relative.

6. Acknowledgments

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

7. References

[1] Brecht Desplanques, Kris Demuynck, and Jean-Pierre Martens, "Factor analysis for speaker segmentation and improved speaker diarization," in *Proc. Interspeech*, 2015, pp. 3081–3085.

[2] Scott Shaobing 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.

[3] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] Luciana Ferrer, Mitchell McLaren, Nicolas Scheffer, Yun Lei, Martin Graciarena, and Vikramjit Mitra, "A noise-robust system for NIST 2012 speaker recognition evaluation.," in *Proc. Interspeech*, 2013, pp. 1981–1985.

[5] Luciana Ferrer, Mitchell McLaren, Aaron Lawson, and Graciarena Martin, "Mitigating the effects of non-stationary unseen noises on language recognition performance," in *Proc. Interspeech*, 2015, pp. 3446–3450.

[6] An Vandecatseye, Jean-Pierre Martens, Joao Neto, Hugo Meinedo, Carmen Garcia-Mateo, Javier Dieguez, France Mihelic, Janez Zibert, Jan Nouza, Petr David, Matus Pleva, Anton Cizmar, Harris Papageorgiou, and Christina Alexandris, "The COST278 pan-European broadcast news database," in *Proc. LREC*, 2004, pp. 873–876.

[7] Brecht Desplanques, Kris Demuynck, and Jean-Pierre 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.

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

[9] Fabio Castaldo, Daniele Colibro, Emanuele Dalmasso, Pietro Laface, and Claudio Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *Proc. ICASSP*, 2008, pp. 4133–4136.

[10] Ondřej Glembek, Lukáš Burget, Pavel Matějka, Martin Karafiát, and Patrick Kenny, "Simplification and optimization of i-vector extraction," in *ICASSP*, 2011, pp. 4516–4519.

[11] Lukáš Burget, Pavel Matějka, Petr Schwarz, Ondřej Glembek, and Jan 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.

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

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

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

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

[16] Mitchell McLaren, Martin Graciarena, and Yun Lei, "Softsad: Integrated frame-based speech confidence for speaker recognition," in *Proc. ICASSP*, 2015, pp. 4694–4698.

[17] Brecht Desplanques and Jean-Pierre Martens, "Model-based speech/non-speech segmentation of a heterogeneous multilingual TV broadcast collection," in *International Symposium on Intelligent Signal Processing and Communication Systems, Proceedings*, 2013, pp. 55–60.

[18] 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`.