Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

Online Two Speaker Diarization

Hagai Aronowitz[1], Yosef A. Solewicz[2], Orith Toledo-Ronen[1]

[1]IBM Research – Haifa, Haifa, Israel
[2] Technology Section, Israel National Police, Jerusalem, Israel
{hagaia, oritht}@il.ibm.com, solewicz@police.gov.il

Abstract

Short conversations pose some challenges for online diarization due to data sparseness and unbalanced representation of the two speakers. This paper presents our recent advances in online diarization of two-wire telephone conversations, introducing several methods for improving processing efficiency and accuracy on short conversations. Our framework is based on the offline diarization of a conversation prefix followed by an efficient online processing of the rest of the conversation. We use an adaptive prefix size, resulting from the tradeoff between desired efficiency and accuracy as measured by a confidence measure on the diarization output. We further show the enhancement of our online speaker recognition system based on implicit speaker diarization using the proposed techniques.

1. Introduction

Speaker diarization is the task of "who spoke when". This paper is part of our ongoing [1-5] work on speaker diarization in summed (two-wire) telephone conversations which are mostly two-speaker based. Our work is motivated by a requirement for an accurate, robust, efficient and online diarization solution which can be either used as a preprocessing phase for speaker verification and speech recognition systems or may be used as an information source for speech analytics systems. The typical use cases are for law enforcement and for contact centers which often have access to summed data only.

Lately [3] we have proposed a novel method for two-speaker diarization based on supervector parameterization of short audio segments, unsupervised (session based) intra speaker within-session variability modeling, and PCA (Principal Component Analysis) based clustering. The proposed method was reported to have good accuracy, reasonable efficiency and does not require any training data. However, it has two shortcomings. First, the method is essentially an offline method. Second, the method was tested on standard five minutes telephone conversations. Recently, while evaluating the method on realistic contact center data, we observed a significant degradation in accuracy when processing short conversations which happen to be quite frequent in real data. Other state-of-the-art methods for speaker diarization in summed telephone conversations [6] also have shortcomings such as the need for a huge developments set, offline computation and unreported results for short conversations.

Speaker diarization in short conversations has been investigated in [7] where sessions as short as 100 seconds were evaluated. However, in our work we focus on much shorter lengths (starting from 15 seconds) which raise different issues than in [7].

Online speaker diarization has been investigated in several works such as [8-11]. In [8, 10], online speaker diarization in European parliament plenary speeches [8] and broadcast news [10] was performed relying on accurate detection of new speakers on-the-fly and using speaker models that were trained according to online decisions, an approach which may lead to error accumulation. In [9], online speaker diarization in meetings was achieved by the use of hybrid online/offline processing which makes use of all available information to train speaker models (and not relying completely on online decisions), thus avoiding error accumulation. However, the underlying speaker diarization technology used in [9] is very different from the technology we use in [3] which achieves very accurate speaker diarization in summed telephone conversations.

Contrary to [8-10], the work in [11] does address the task of speaker diarization in summed telephone conversations. The approach taken in [11] is to run the offline diarization system on a prefix of the conversation. The outcome of the prefix processing is a segmentation of the prefix and trained acoustic models for each hypothesized speaker. The rest of the conversation is simply decoded using the trained acoustic speaker models using Viterbi decoding. The method we take in this paper approach extends the approach described in [11].

This paper reports our efforts for improving our speaker diarization method to be efficient, online and robust to conversation length. The contributions of this paper are as follows. We introduce the concept of intra-speaker within-session variability modeling from an unlabeled development set consisting of summed conversation.

Next, we introduce the concept of outlier-emphasizing-PCA that gives a larger weight to outlier vectors in the PCA analysis. Outlier-emphasizing-PCA enables PCA-based clustering to better cope with short sessions for which very often one of the speakers is underrepresented in the session. Furthermore we modified our probabilistic model derived in [3] to be more robust to underrepresented speakers.

Our basic framework is similar to the one in [11] in the sense that we separate a conversation into a prefix that we process in an offline manner, and then use Viterbi to decode the rest of the conversation using trained speaker models from the prefix processing. However, we set the prefix adaptively using a confidence measure estimated on-line. The adaptive prefix length approach enables us to use in general short prefixes, and use larger prefixed only when required. Furthermore, contrary to [11] we do update our speaker models periodically after prefix processing in order to get a more accurate diarization.

Last but not least, we show how the proposed techniques significantly enhance our online two-wire recognition system based on implicit speaker diarization [4].

The remainder of this paper is organized as follows: Section 2 describes the baseline supervector-based speaker diarization system. In Section 3 we describe our methods for obtaining improved speaker diarization accuracy in short conversations. In Section 4 we describe our proposed online speaker diarization system. In Section 5 we present our enhanced online implicit speaker diarization system for speaker verification in summed telephone conversations. In Section 6 we describe the experimental setup, datasets and results. Finally, we conclude in Section 7.

2. Supervector-based speaker diarization

Our baseline speaker diarization system is based on unsupervised compensation of intra-speaker within-session variability followed by PCA-based clustering and is described in detail in [3]. The system parameterizes the audio using GMM (Gaussian Mixture Model) supervectors extracted for evenly overlapping one-second superframes. Intra-speaker within-session variability is estimated in an unsupervised manner and removed from the GMM-supervectors using the NAP (Nuisance Attribute Projection) method. The system exploits the assumption that only two speakers are expected in a session. This assumption is used by applying PCA to the compensated GMM-supervectors scatter matrix and distinguishing between the two speakers by taking each GMM-supervector and classifying it according to the sign of its projection on the *largest* eigenvector (the one corresponding to the largest eigenvalue). The segmentation is smoothed using Viterbi decoding, and refined by applying Viterbi re-segmentation using the original frame-based features. The system achieves a SER (speaker error rate) of 2.8% on NIST 2005 summed telephone conversations.

In the following subsections we describe in more detail the different components of the system.

2.1. Front-end

The front-end used in our system is based on Mel-frequency Cepstrum coefficients (MFCC). The feature set consists of 13 cepstral coefficients extracted every 10 ms using a 25 ms window. An adaptive energy based voice activity detector (VAD) with Viterbi smoothing is used to locate and remove non-speech frames. The adaptive VAD is essentially offline because it uses an energy histogram calculated from the entire conversation.

2.2. Session-dependent UBM

A GMM-UBM (Universal Background Model) is estimated independently for each session. This approach eliminates the need of a development set. However, it is inherently offline.

2.3. Supervector parameterization

We parameterize the speech signal with a time series of supervectors. The speech signal is divided into evenly spaced overlapping superframes (sequences of frames) of one second length with an offset of 100 ms (superframe rate is 10/second). We estimate a supervector for each superframe using standard MAP (Maximum a Posteriori) adaptation. The parameterization procedure is outlined as following:

GMM-supervector parameterization

1. Define evenly spaced overlapping superframes of one second length with an offset of 100 ms.
2. Estimate a GMM for each superframe by adapting the UBM to the frames of the superframe using standard MAP.
3. Parameterize each superframe with the supervector created by concatenating the means of its estimated GMM.

2.4. Intra-speaker within-session variability compensation

We assume that most of the intra-speaker within-session variability is confined to a low dimensional affine subspace in the supervector space. In order to estimate this subspace without any development data, we estimate it independently for each conversation. This is done by exploiting the fact that speaker turns are in general longer than the one-second superframes we use to parameterize the speech signal. We can therefore assume that pairs of overlapping adjacent segments usually belong to the same speaker. According to this assumption, we estimate the covariance matrix of the difference supervectors between adjacent overlapping one second segments and use PCA to estimate the intra-speaker subspace. The estimated subspace is removed from all supervectors using the NAP method. Intra-speaker within-session variability compensation gave a 40% SER reduction (4.8%→2.8%) in our experiments [3]. Regarding online processing, our method for intra-speaker variability estimation is inherently offline.

2.5. PCA-based supervector clustering

After intra-speaker variability compensation, we assume that most of the supervector variability is accounted to speaker identity. We use the following recipe to cluster the audio into two clusters:

1. Compute the covariance matrix of the compensated supervectors.
2. Apply PCA to find the *largest* eigenvector.
3. Project each compensated supervector onto the *largest* eigenvector.
4. Use the projections for estimating an LLR (log-likelihood ratio) for each superframe with respect to the two speakers (more details in subsection 3.4 below).
5. Viterbi segmentation is used to create a smoothed segmentation from the superframe-based LLRs.

Regarding online processing, both steps 2 and 5 are offline in nature.

2.6. Viterbi re-segmentation

The segmentation obtained from the PCA-based supervector clustering is used to train a GMM for each speaker using the original frame-based feature vectors. The GMMs are used by a Viterbi decoder to produce a refined segmentation. The adaptation-segmentation scheme is iterated for several iterations (two in our setup).

3. Robustness to short sessions

Achieving accurate diarization for short sessions is essential for an offline diarization system dealing with realistic data, and is also an important step in a development of an online system. In addition to the problem of data sparseness which we address in subsections 3.1-3.2, short sessions are more likely to suffer from underrepresented speakers, a problem we directly address subsections 3.3-3.4.

3.1. Offline UBM & NAP training

In our baseline system the UBM and the NAP projection are trained on-the-fly for every session independently. In short sessions, the available data may be insufficient. We therefore propose to train the UBM and the NAP projection from an unlabeled development set consisting of summed conversations.

The training of a UBM from an unlabeled development set is straightforward. The NAP projection is trained by processing the development set with the steps described in subsections 2.1-2.4 and pooling the individual covariance matrices (used to train session-dependent NAP) over the entire development set. The offline-trained NAP projection is obtained by applying PCA on the pooled covariance matrix.

3.2. Reduced GMM orders

In [3] the GMM orders were optimized to maximize accuracy on five-minute sessions, resulting with 64 Gaussians for the UBM and 32 Gaussians for the final speaker GMM models. However, the optimal GMM orders should be lower when sessions are short, and in our setup for the proposed system we set the GMM orders to 16 Gaussians for both the UBM and the speaker models.

3.3. Outlier-emphasizing PCA

The accuracy of the segmentations obtained by our diarization system is heavily dependent on the assumption that the dominant component in the supervector scatter matrix is the speaker identity. However, in cases of an underrepresented speaker, the supervector scatter matrix may be dominated by the dominant speaker's intra-speaker variability.

In order to increase the influence of an underrepresented speaker in the supervector scatter matrix, we exploit the fact that the supervectors of such a speaker may be considered as outliers compared to the supervector sample. We can therefore replace the supervector scatter matrix with a weighted scatter matrix and assign large weights to supervectors which we consider as outliers and smaller weights to supervectors we consider inliers. In this work we detect outliers by selecting the top 10% supervectors in a given session with the largest distance to the sample mean.

3.4. Adaptive LLR calibration

In [3] we show that the LLR of a compensated superframe c_i with respect to speakers s_1 and s_2 is a linear function of the projection of the corresponding supervector on the *largest* eigenvector (p_i):

$$\log\frac{\Pr(c_i|s_1)}{\Pr(c_i|s_2)} \approx ap_i + b \cdot \quad (1)$$

In [3] we assumed that both speakers are equally represented in the call. This assumption led to setting the bias b to zero. We further assumed that a is session independent. In order to be robust to unequal representation of speakers, we estimate the bias b by averaging the 10% and 90% percentiles of the projected supervectors sample.

4. Online speaker diarization

4.1. Online VAD

In order to decrease the latency of the VAD subsystem we limit the estimation of the energy histogram to a predefined prefix of the conversation denoted by P_v. For a required latency L_v, the Viterbi decoding (used for smoothing) is replaced by the following online version:

Online Viterbi decoding
For every frame t:
1. Compute forward Viterbi probabilities for frame t.
2. If $t = nL_v/2$ for n=2,3,…
   Backtrack from frame t to frame t-L_v and report state sequence for frames [t-L_v+1 ,…,t-L_v/2]

In this work we use P_v=15 seconds and L_v=0.1 seconds. This setup ensures that the latency of the diarization system is hardly affected by the latency of the VAD component.

4.2. Online PCA-based supervector clustering with Viterbi re-segmentation

In order to convert the offline sequence of PCA computation, Viterbi smoothing and iterative Viterbi re-segmentation to an online setting, we define a prefix of the conversation for which the processing is done offline. The result of this step is a segmentation reported for the prefix, an estimated *largest* eigenvector, and trained speaker models.

After processing the initial prefix, the trained speaker models are used to run an online Viterbi decoder as described in subsection 4.1 to obtain a segmentation with a predefined latency. In order to benefit from online accumulated data, the PCA and Viterbi re-segmentation steps are re-executed periodically using the accumulated speech signal. Following is an outline of the online algorithm with a prefix denoted by P_d, a retraining period denoted by R_d, and delay denoted by L_d. Note that step 2.D is essential for preventing the PCA algorithm to obtain inconsistent clustering when it is re-executed with more data.

Online speaker diarization
For every frame t
1. Accumulate statistics for PCA calculation
2. If t=P_d+nR_d for n=0,1,2,…
   A. Compute PCA using accumulated statistics (for frame 1,…,t)
   B. Calculate LLRs for frames 1,…,t
   C. Calculate a smooth Viterbi segmentation for [1,…,t]
   D. if $n>0$
      Verify that the new segmentation is consistent with the previous segmentation. If not, swap the first speaker with the second speaker.
   E. Iterate Viterbi re-segmentation to obtain a refined

segmentation and trained speaker models

2. If $t=P_d$
   Report segmentation for prefix [1,…,t-L_d]
3. Use the most updated speaker models to calculate an LLR for frame t.
4. If $t = P_d+nL_d/2$ for n=1,2,…
   Backtrack from frame t to frame t-L_d and report segmentation for frames [t-L_d+1 ,…,t-L_d/2]

4.3. Confidence-based prefix length

The proposed online system is useful only if the prefix (P_d) is short enough. However, for short prefixes many conversations contain significant speech from only a single speaker. For instance, on the NIST-2005 dataset that we use for our experiments in Section 6, for a 15 seconds prefix, only 60% of the sessions contain two speakers with more than one second of speech, and only 35% of the sessions contain two speakers with more than three seconds of speech. Even for a 30 seconds prefix, only about 70% of the sessions contain more than three seconds for two speakers.

The strategy we propose is as follows. For a given session, we start by setting a short prefix (15 seconds). After obtaining the segmentation for the prefix, a confidence measure is estimated. In case of a low confidence, we extend the prefix to a longer duration (30 seconds), etc. Using a clustering validity measure as a confidence measure of the diarization quality on the leading prefix of the call, we can either reduce the latency of the diarization process if the confidence is high, or prolong the prefix as needed if the confidence is low. We apply the clustering validity measure to the segmentation output obtained from diarization of the prefix segment and use this measure for adaptive selection of the prefix size.

We explored several well-known clustering validation algorithms for guiding the online diarization process including the Davies-Bouldin (DB) validation index [12] and the Silhouette validation method [13]. We present our confidence-based diarization results with the DB validation index, which gave us the best results. The DB index provides an overall score for the entire segmentation. It is defined as the ratio between the sum of the two standard deviations of the data points' distances in each cluster to their cluster center and the Euclidian distance between the two clusters centers. The Silhouette method will be used for the speaker recognition experiments in Section 5. The Silhouette is a measure of the similarity of a data element to the elements in its cluster compared to the elements in other clusters. This measure provides a score for every superframe in segmentation.

5. Online implicit speaker diarization for speaker verification

Speaker diarization is particularly useful as a preprocessing phase for speaker recognition in summed conversations. Focusing on this particular use case, we took in mind in [4] that the optimization criterion in such a setup is minimization of speaker recognition error rather than minimization of speaker diarization error rate (SER). We therefore proposed in [4] to use a framework that integrates an implicit speaker diarization step into the speaker recognition process. The basic algorithmic ideas were inspired by our speaker diarization work in [3], but the different optimization criterion and time complexity requirements led to some significant modifications.

In short, the method in [4] works by dividing the audio into overlapping five-second superframes and scoring independently each superframe against a target speaker model. A partial diarization processing is used to cluster the superframes into two clusters and discard superframes which are in the borderline between the two clusters.

The superframe classification process was implemented in [4] by using the PCA-based supervector clustering method described in subsection 2.5 with the exception of using a simple thresholding instead of using Viterbi smoothing. In order to achieve low latency, the PCA was done only on a prefix of the conversation. However, it was observed that a minimal prefix of 60 seconds was required to obtain good results.

Our current goal is to reduce the required prefix significantly. We present in the following subsections a better superframe classification scheme leading to substantial reduction in the latency requirements towards online diarization.

5.1. Eigenvoice-based diarization

Our proposed method attempts to overcome the delay caused by PCA training from scratch by incorporating prior information into the learning process. This is accomplished by replacing the conversation-dependent PCA projection with a fixed subspace projection spanning the most informative speaker directions. This unique base is actually the low-rank representation of the speaker space, also known as Eigenvoices [14]. Eigenvoice-based diarization decreases the relative large amount of learning data required by the PCA method at the expense of a larger projection subspace rank. The augmented projection space demands more sophisticated superframe classification schemes than simple thresholding used for the one-dimensional PCA projection. In the current implementation, k-means clustering is used to classify incoming superframes into two speakers. A related approach is described in [15], where clustering is used on the main PCA projections of the total variability space for diarization. Although conceptually similar to our approach, preliminary experiments suggested that apart from diarization, recognition performed on the clustered low-rank features (either Eigenvoices or the total variability) performs poorer than if performed on the non-projected superframes, as we propose.

5.2. Clustering quality

Our methodology can be further improved by considering the concept of clustering validation discussed in 4.3. Clustering quality measures can help us determining how well the speakers' centroids can be determined given a certain prefix of the conversation. Besides enabling us to dynamically optimize the prefixes' length, these measures can be used to enhance the recognition process by spotting unseen outlier superframes or even foreseeing difficult conversations.

Actually, the noise-floor threshold for discarding borderline superframes used with the one-dimensional PCA diarization [4] can be seen as a simple clustering quality mechanism. A drawback of that method is that noise

thresholds determined based on the PCA training prefix are not stable for unseen superframes.

6. Experiments and results

6.1. Datasets and protocol for the speaker diarization experiments

A subset of the NIST-2005 SRE core dataset was used as a development set (131 sessions), and a disjoint subset was used as an evaluation set (916 sessions). We artificially convert the stereo datasets to mono by summing both channels. The ground truth was derived from the automatically produced transcripts provided by NIST.

Speech/non-speech segmentation is not the main focus of this work. Therefore, we use the standard speaker error rate (SER) measure and do not include speech/non-speech errors. SER is computed according to the standard protocol for evaluation of a two-speaker segmentation task, which is available from NIST [16]. However, in order to improve our assessments in short sessions, we do not discard the margins around speaker turns (for all our results), as done in [16]. We therefore report slightly degraded accuracy compared to what we would have obtained by discarding the margins.

For the sake of the analysis of short sessions, we have cut the original NIST five-minute sessions into short sessions of variable lengths by taking the prefixes of each session. In our preliminary experiments we realized that a significant amount of these short prefixes do not have an adequate representation of two speakers. We therefore limit the analysis to prefixes which contain at least three seconds of net speech per speaker.

6.2. Short sessions experiments – selected results

Table 1 presents results for our baseline system [3] and our proposed system which included all the capabilities described in Section 3.

Table 1: SER for the proposed system compared to the baseline as a function of session length.

Session length (in seconds)	Baseline system SER (in %)	Proposed system SER (in %)	Relative improvement (in %)
15	22.2	9.9	55
30	17.6	8.8	50
60	13.3	7.6	43
90	10.2	6.7	34
120	7.9	5.6	29
240	5.0	4.6	8
300	4.4	4.4	0

The results in Table 1 show a clear superiority of the proposed system compared to the baseline system, especially for short sessions.

6.3. Short sessions experiments – detailed results

Figure 1 presents detailed results for short session experiments. Note that the results for 15 seconds sessions are generally better than those for 30 seconds sessions due to the removal of sessions with less than three seconds of speech per speaker, which prevents the 15 seconds sessions to have an extreme speaker imbalance. In fact, we found out in our experiments that the degree of imbalance between speakers is the most important factor for predicting the accuracy of our system.

Analyzing the results we can conclude that all systems perform roughly the same for long sessions (240 and 300 seconds). For short sessions, we see that in general all our proposed variants improve performance and combine favorably.

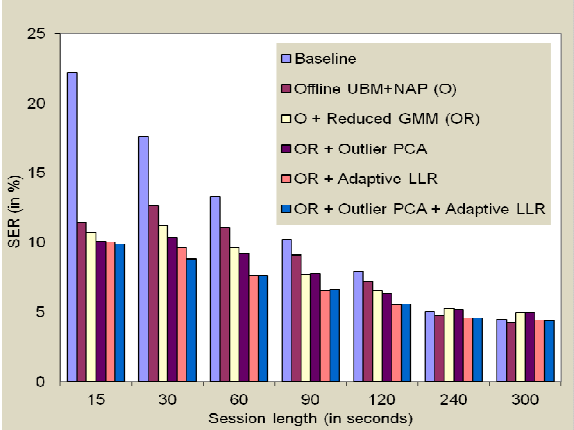

Figure 1: SER for short sessions using different algorithmic variants described in Section 3.

6.4. Online diarization results

A tradeoff between the latency, accuracy and time complexity of the online system can be achieved by controlling the VAD parameters (prefix length and delay) and the speaker diarization parameters (prefix length, retraining period and delay). The VAD parameters were discussed in subsection 4.1. Regarding the speaker diarization parameters, we set the retraining parameter (which is unrelated to the latency) to 15 seconds, and report results for various delay values and various prefix values.

6.4.1. Speaker diarization delay parameter

Table 2 presents online diarization results as a function of the delay parameter for prefixes of 15 and 30 seconds. The results in Table 2 indicate that a delay parameter value of 0.2 is optimal.

Table 2: SER for the online diarization system as a function of the delay parameter.

Delay (in seconds)	0.1	0.2	0.5	1
Prefix = 15 seconds SER (in %)	9.6	9.0	9.0	9.0
Prefix = 30 seconds SER (in %)	8.3	7.7	7.7	7.7

6.4.2. Speaker diarization prefix parameter

Table 3 presents online diarization results as a function of the prefix parameter (the delay parameter is set to 0.2 seconds). The second row presents results for all sessions that pass the three seconds net speech per speaker criterion. The third row

presents results for all sessions for which the prefix passes the same criterion. A comparison between the two rows indicates that our online diarization system is able to deal very well even with short prefixes as long as the two speakers are reasonable represented in the prefix.

Table 3: SER (in %) for the online diarization system as a function of the prefix parameter. The delay parameter is set to 0.2 seconds.

Prefix (in seconds)	15	30	45	60	90	120	offline
3 seconds per speaker in session	9.0	7.7	6.8	6.2	5.4	4.9	4.4
3 seconds per speaker in prefix	6.4	5.7	5.6	5.5	5.1	4.8	4.4

6.4.3. Latency

The latency of the proposed online diarization system is P_d (prefix length) for the prefix. For the rest of the conversation, it is the sum of the VAD delay, the diarization delay and the superframe length: L_v+L_d+1, which is 1.3 seconds for the configuration found to be optimal in subsection 6.4.1.

6.5. Confidence-based scoring for diarization

The confidence score on the call prefix can be useful for determining if the segmentation accuracy is good enough for a given prefix or we need to prolong the prefix. At a given prefix length, we compute the confidence on the prefix and select some percentage of the calls with the highest confidence. For the selected calls we make a decision to use the given prefix for the diarization, and the performance on that subset of calls is the online diarization with the corresponding prefix. For the rest of the calls with the lower confidence, we use a longer prefix, and the performance for that subset is the online diarization with a longer prefix. The overall accuracy on all the calls is computed by the combination of the two subsets.

We demonstrate this approach by showing the confidence results for two prefix lengths: 15 and 30 seconds. We present the results obtained with the DB index method described in subsection 4.3. Figure 2 presents the overall performance results for four different combinations of prefix lengths. The legend P1-P2 means a combination of a short prefix P1 in seconds and a long prefix P2. The P1 prefix is used for confidence computation of all the calls and for diarization of a selected subset of calls with the highest confidence, while prefix P2 is used for diarization of the rest of the calls. The x-axis represents the percentage of the calls selected by the confidence scoring performed on the shorter prefix P1. This means that the accuracy value at x=1 represents the online diarization accuracy with the shorter prefix P1, while the value at x=0 is the accuracy with the longer prefix P2. The data points along each line represent different combinations of the two prefix sizes. These results show that when selecting up to 20% of the calls with the highest confidence for a 15-seconds prefix, the accuracy is preserved. With a longer prefix of 30-seconds for confidence computation, the accuracy is preserved at a wider range up to 50%.

Figure 3 presents the performance as a function of the mean latency achieved by each confidence-based combination of a short and a long prefix (P1-P2) using the DB Index confidence score. We can see, for example, that for a mean prefix of 45 seconds, the best performance is achieved by the 30-60 combination with SER=6.41% for a combination of 50% with P1=30 seconds and 50% with P2=60 seconds, while the overall performance of a single prefix with P1=45 seconds is only SER=6.84%. This is an indication that confidence scoring is useful for improving the performance of the online diarization process.

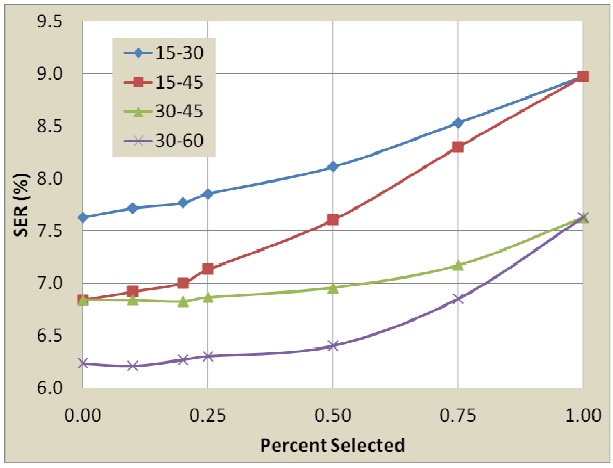

Figure 2: Performance of confidence-based diarization of short and long prefixes combinations as a function of the percentage of calls selected by the confidence.

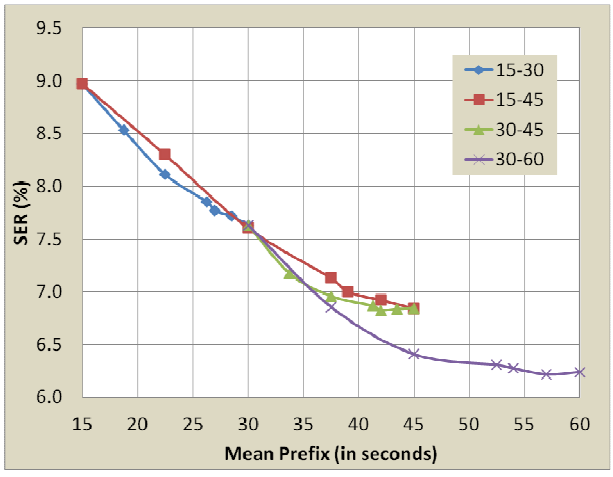

Figure 3: Performance of confidence-based diarization of short and long prefixes combinations as function of the mean prefix length.

6.6. Speed analysis

Table 4 presents an analysis for selected offline and online diarization systems described in this paper. For each system the accuracy and speed are reported. The analysis is done for

full (five minutes) conversations. No sort of optimization was done for the C++ implementation of the diarization systems.

Under the offline framework, the best accuracy (SER of 3.5%) is achieved using our baseline system with two modifications (outlier PCA, adaptive LLR). However, this system runs only 5 times faster than RT. The proposed offline system (Section 3 and subsection 6.2) is much more efficient (50 times faster than RT) with some reduced accuracy (4.4%). Moreover, as shown in subsection 6.2 the proposed system copes much better with short conversations.

Under the online framework, a tradeoff between accuracy and speed is reported in Table 4. Basically the speed is controlled by the frequency of PCA and GMM retraining (retrain parameter). A good tradeoff may be obtained by varying the retraining frequency in such a way that at the beginning of the conversation the frequency of retraining is higher compared to the frequency later on in the conversations (var. #1 and var. #2).

Table 4: Speed and accuracy analysis of selected offline and online diarization systems.

System	SER (in %)	Speed (xRT)
Offline Diarization		
Baseline [3]	4.4	5
Baseline + Outlier PCA + Adaptive LLR	3.5	5
Proposed system (subsection 6.2)	4.4	50
Online diarization, prefix=30, delay=0.3		
Retrain parameter = 15	7.7	17
Retrain parameter = 30	8.0	26
Retrain parameter = 60	8.9	36
Retrain parameter – var. (#1)	7.8	30
Retrain parameter – var. (#2)	8.0	40

6.7. Datasets and protocol for speaker verification experiments

The speaker recognition experiments were performed on the male subset of the NIST-2005 SRE core dataset. Speaker models are trained using the four-wire conversations defined by the NIST protocol. Two-wire testing sessions were obtained as before by artificially summing the two sides of the testing conversations in the original protocol. At all, there are 274 speaker models, and around 950 and 8000 target and impostor trials respectively. Data from NIST 2004 and 2006 campaigns is used for UBM, background modeling, NAP and Eigenvoices estimation.

6.8. Eigenvoice vs. PCA diarization

The following experiments compare the effectiveness of the proposed Eigenvoice based diarization in Section 5 with the former PCA approach [4]. We use the same GMM-NAP-SVM used in [4] and briefly described above. In particular, we investigate optimum subspace ranks, delays involved and the usefulness of the clustering quality measure.

The superframe sequence is classified into two groups either by thresholding their PCA projection (as in the original framework) or by k-means clustering of their Eigenvoice decomposition. Both (non-projected) superframe groups are scored against the speaker model and the highest of each group's average score is the ultimate recognition score. Note that the experiments focus on the on-line diarization capabilities of the proposed method, although recognition performance is evaluated for the entire conversations.

Initially, we show in Table 5 the impact of the Eigenvoice (EV) space rank on recognition performance. For comparison, we show the PCA-based method including higher PCA dimensions beyond the main axis used originally. For both cases, k-means is used for superframe classification. We observe that a few Eigenvoice directions are equivalent to the main eigenvector estimated per conversation. Moreover, no gains are obtained by increasing the dimension of the PCA rank.

The next experiment investigates the performance as a function of the prefix length, as seen in Figure 4. Distinct prefix lengths are used for estimating the main PCA axis and the Eigenvoice (rank=25) clusters. We present two additional versions of the Eigenvector technique incorporating clustering quality in the recognition process. Among the various existing clustering quality measures mentioned in Section 4.3, we use in these experiments the Silhouette method. Superframe silhouettes are calculated given the estimated centroids for different prefixes. The Silhouette originally ranges from -1 to +1, although we re-normalized this range to [0,1]. Therefore, a superframe with assigned silhouette of zero is probably an outlier, while a silhouette value of one represents a well clustered superframe. The first Eigenvector version simply discards superframes possessing silhouettes less than 0.5 before scoring (EV_h). The second version weights each supervector by its silhouette value during scoring (EV_s). The experiment confirms that the Eigenvector method clearly outperforms the former PCA approach in reducing training delays. In addition, we observe that the incorporation of clustering quality measures further enhance the Eigenvector method, especially for low prefixes as could be expected.

We finally investigate the use of variable prefix lengths within Eigenvoice diarization. In this experiment, we progressively check 10, 20 and 30 second-prefixes, until we reach some target quality for the prefix clustering. Figure 5 shows recognition performance for the several methods as a function of the targeted silhouette value. Higher quality values will improve performance at the expense of larger prefixes. The corresponding average prefix length across the whole evaluation is shown for each silhouette. The results obtained support the idea of using variable prefix lengths for distinct conversations. In average, variable prefixes roughly halves the delay introduced with fixed prefixes.

Table 5: DCF (x100) as a function of the rank for PCA and Eigenvoices projections.

Rank	1	5	10	25	50
PCA	2.30	2.68	2.96	2.71	2.65
EV	2.52	2.30	2.28	2.19	2.17

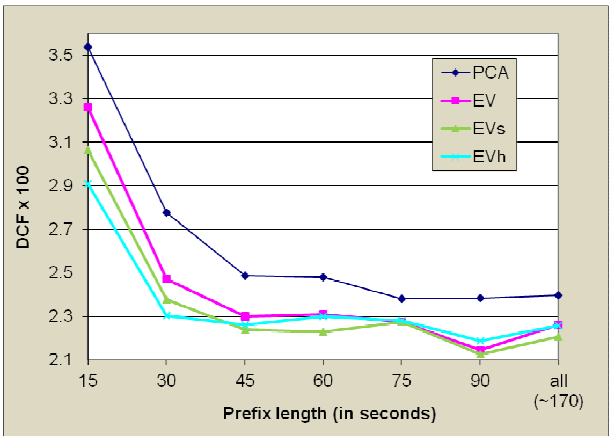

Figure 4: Performance as a function of the prefix length for the PCA and Eigenvector projections.

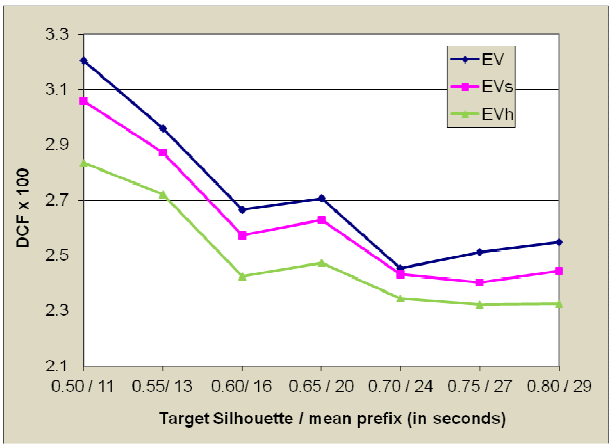

Figure 5: Performance as a function of the target silhouette and the correspondent average prefix length.

7. Conclusions

In this paper we have extended our recently developed method for speaker diarization [3] to cope with short conversations, and to perform online diarization. For coping with short and frequently speaker-unbalanced conversations we proposed the following novelties: offline unsupervised estimation of intra-session intra-speaker variability, outlier emphasizing PCA for improved speaker clustering and adaptive calibration of speaker log likelihood ratio calibration. Our proposed online diarization system builds on the novelties discussed above and achieves a low latency (1.3 seconds) except for a prefix of variable length (15-60 seconds) which we determine according to a confidence measure. By setting the length of the prefix adaptively we manage to reduce the expected prefix length by 25% with a very small degradation in accuracy. In order to obtain improved accuracy we redo the PCA analysis and clustering and retrain the speaker models periodically during the online processing.

In terms of speed, our proposed offline system runs 50 times faster than real-time without any code optimization. The proposed online system runs 30-40 times faster than real-time. Finally, substantial delay reduction was also achieved on our summed-channel speaker recognition system. Diarization performed on the Eigenvector instead of PCA domain halved original delay requirements. The average delay can be further halved by using our concept of variable prefixes.

8. References

[1] Aronowitz, H. and Solewicz, Y., "Speaker Recognition in Two Wire Test Sessions," in Proc. *Interspeech*, 2008.

[2] Solewicz, Y.A., Aronowitz, H., "Two-Wire Nuisance Attribute Projection", in Proc. *Interspeech* 2009.

[3] Aronowitz, H., "Unsupervised Compensation of Intra-Session Intra-Speaker Variability for Speaker Diarization", in Proc. *Speaker Odyssey*, 2010.

[4] Solewicz, Y., Aronowitz H., "Implicit Segmentation in Two-Wire Speaker Recognition", in Proc. *Interspeech*, 2011.

[5] Aronowitz, H., "Speaker Diarization using A Priori Acoustic Information", in Proc. *Interspeech*, 2011.

[6] Reynolds, D., Kenny, P., and Castaldo, F., "A Study of New Approaches to Speaker Diarization", in Proc. *Interspeech*, 2009.

[7] Imseng, D., and Friedland, G., "Robust speaker diarization for short speech recordings", in Proc. *ASRU*, 2009.

[8] Markov, K., Nakamura, S., "Never-ending learning system for on-line speaker diarization", in Proc. *ASRU*, 2007.

[9] Vaquero, C., Vinyals, O., Friedland, G.,"A hybrid Approach to Online Speaker Diarization", in Proc. *Interspeech*, 2010.

[10] Geiger, J., Wallhoff, F. and Rigoll, G.,"GMM-UBM based open-set online speaker diarization", in Proc. *Interspeech*, 2010.

[11] Ben-Harush, O., Lapidot, I., Guterman, H., "Online Diarization of Telephone Conversations", in Proc. *Speaker Odyssey*, 2010.

[12] D.L. Davies, D.W. Bouldin, "A cluster separation measure", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 2, 1979, pp. 224-227.

[13] P.J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", *Journal of Computational and Applied Mathematics*, Vol. 20, 1987, pp. 53-65.

[14] Thyes, O., Kuhn, R., Nguyen, P., and Junqua, J.-C., "Speaker identification and verification using eigenvoices", in Proc. *ICSLP*, 2000.

[15] Shum S., Dehak N., Chuangsuwanich E., Reynolds D., and Glass J., "Exploiting Intra-Conversation Variability for Speaker Diarization," in Proc. *Interspeech*, 2011.

[16] NIST segmentation scoring script, Available online: "http://www.itl.nist.gov/iad/mig/tests/sre/2002/SpkrSegEval-v07.pl", 2002.