Odyssey 2010
The Speaker and Language Recognition Workshop
28 June – 1 July 2010, Brno, Czech Republic

Online Diarization of Telephone Conversations

Oshry Ben-Harush†, *Itshak Lapidot*°, *Hugo Guterman*†

† Department of Electrical and Computers Engineering
Ben-Gurion University of the Negev, Beer-Sheva, Israel
○ Department of Electrical and Electronics Engineering
Sami Shamoon College of Engineering, Ashdod, Israel
oshryb@bgu.ac.il, itshakl@sce.ac.il, hugo@ee.bgu.ac.il

Abstract

Speaker diarization systems attempts to perform segmentation and labeling of a conversation between R speakers, while no prior information is given regarding the conversation. Diarization systems basically tries to answer the question "Who spoke when?".

In order to perform speaker diarization, most state of the art diarization systems operate in an off-line mode, that is, all of the samples of the audio stream are required prior to the application of the diarization algorithm. Off-line diarization algorithms generally relies on a dendogram or hierarchical clustering approach.

Several on-line diarization systems has been previously suggested, however, most require some prior information or off-line trained speaker and background models in order to conduct all or part of the diarization process.

A new two-stage on-line diarization of telephone conversations algorithm is suggested in this study. On the first stage, a fully unsupervised diarization algorithm is applied over an initial training set of the conversation, this stage generates the speakers and non-speech models and tunes a hyper-state Hidden Markov Model (HMM) to be used on the second, on-line stage of diarization.

On-line diarization is then applied by means of time-series clustering using the Viterbi dynamic programming algorithm. Employing this approach provides diarization results a few miliseconds following either a user request or once the conversation has concluded.

In order to evaluate diarization performance , the diarization system was applied over 2048, 5Min length, two-speaker conversations extracted from the NIST 2005 Speaker Recognition Evaluation.

On-line Diarization Error Rate (DER) is shown to approaches the "optimal" DER (achieved by applying unsupervised diarization over the entire conversation) as the length of the initial training set increases. Using an initial training set of 2Min and applying on-line diarization to the entire conversation incurred approximately 4% increase in DER compared to the "optimal" DER.

1. Introduction

Given a conversation between R speakers, speaker diarization systems attempts clustering and labeling of temporal conversation segments to $\{S_r\}_{r=1}^{R}$ speakers and to non-speech, while no prior information is given regarding the conversation.

Conversation diarization is essential for several speech processing applications such as, conversation indexing, forensics, automatic speaker modeling and as a pre-processing stage for speaker recognition tasks. Diarization of conversations could also contribute to increased accuracy of Automatic Speech Recognition (ASR), as these systems exhibits improved performance operating on a speaker-dependent mode. Forensics labs require processing of an increasingly large amount of audio data, it could be beneficial to perform an initial review of the required audio stream with an on-line diarization system and apply more accurate audio diarization algorithms once the required audio segment was found.

Major part of state of the art diarization systems operate in an off-line mode, that is, all of the conversation samples must be at hand prior to the application of the diarization algorithm. Diarization is then applied using some hierarchical or dendogram clustering which usually relies on the Bayesian Information Criterion (BIC) [1] for either change detection, cluster recombination or both, e.g., [2, 3, 4].

However, on-line diarization of an audio stream might be imperative for some speech processing applications, such as, continuous or spontaneous speech recognition, speech to text and surveillance applications. These systems generally operate in an on-line, or sometimes in a real-time manner and, thus, requires on-line diarization systems.

Conversation diarization is a complex task to perform while given all of the conversation data a-priori, it becomes exceedingly complicated to accomplish in an on-line manner. Several examples of on-line diarization systems can be found in the literature, Markov and Nakamura [5] suggests on-line GMM learning along with a form of novelty detection in order to assign a new segment to one of the previously generated clusters or to spawn a new cluster, the suggested system requires pre-trained gender and silence models in order to perform speech/non-speech and gender detection. Liu and Kubala [6] used a hybrid approach which integrates a leader-follower approach and a global model selection criterion to perform on-line diarization of broadcast news. Koshinaka et. al. [7] introduces Ergodic Hidden Markov Model (EHMM) and an on-line Expectation Maximization (EM) algorithm for conference meetings diarization.

All of the on-line diarization systems encountered handle multi-speaker conferences or broadcast news scenarios, and most require some labeled data in order to train inherent models used for speech/non-speech classification, gender detection and for spawning speaker models.

Broadcast news and conferences recordings differs greatly from telephone conversations in both environmental and content characteristics. For conferences recordings, environmental conditions are generally familiar, that is, channel conditions are

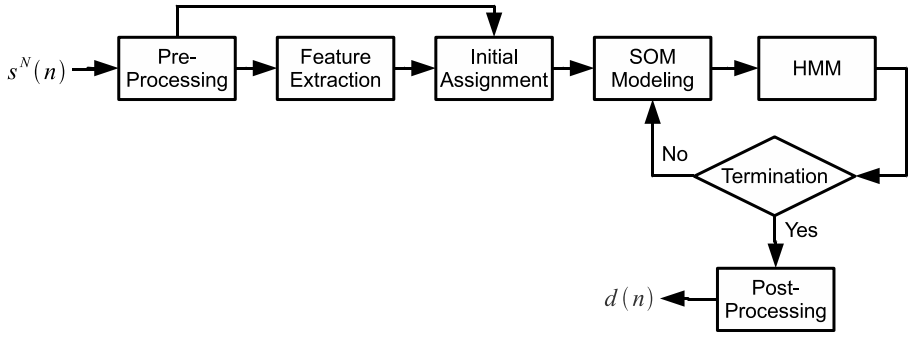

Figure 1: *Baseline diarization system.*

fixed during the entire recording, non-speech segments would generally contain silence. For both broadcast news and conference recordings, the audio stream and speaker turns are generally long in duration. Hierarchical clustering using statistical similarity measures generally exhibits sufficient diarization performance under these conditions, this is due to the abundance in statistics available for model size estimation and for speaker model training.

Telephone conversations, however, does not always share these characteristics, conversations are generally short in duration and speaker turns are usually short, thus, not allowing sufficient statistics to be gathered for the change detection stage of hierarchical clustering [8]. Environmental conditions could change during the conversation, either by changing the channel (recording is made from two different channels) or from a change in scenery (cellular phone recording), thus, speech/non-speech classification task appears much more complex and a-priori training of speaker or background models for speech/non-speech classification seems irrelevant. The use of off-line trained background models for spawning speaker models during the conversation could be implemented, however, channel compensation such as the Nuisance Attribute Projection (NAP) must be applied [9], channel compensation generally requires sufficient amount of speaker data and is appropriate for a unique speaker, and thus, hard to implement.

One major advantage telephone conversation diarization has over conference or broadcast news diarization is that the number of speaker is assumed to be known and fixed ($R = 2$), the diarization system presented would greatly rely on this information.

In this paper, a two-stage, on-line, fully unsupervised telephone conversation speaker diarization system is presented. The suggested on-line diarization system relies on a Self Organizing Map (SOM) based iterative diarization system previously published [10] to perform the first, unsupervised, stage of diarization.

On the first stage, unsupervised diarization is applied over an Initial Training Set (ITS) of the audio stream, this enables the construction of speakers and non-speech models and adaptation of the time-series clustering parameters. The entire audio stream is segmented using the previously trained models by applying the Viterbi dynamic-programming algorithm.

Diarization was applied over 2048 conversation from the NIST 2005 Speaker Recognition Evaluation (SER) [11] using both the baseline system which performs unsupervised diarization given the complete audio stream and the on-line diarization algorithm. Diarization error increases by $\sim 4\%$ compared to the diarization error of the baseline system, this is while using initial training set length of 2Min and applying the on-line diarization algorithm over the entire conversation.

The rest of this paper is as follows: section 2 describes the baseline diarization system including modeling approach, time-series clustering and features required for the unsupervised telephone conversation diarization system. Section 3 introduces the on-line diarization algorithm along with required computational complexity and the on-line diarization methodologies for varying length conversations. Experimentations and results are described in Section 4 and section 5 concludes this study.

2. Baseline Diarization System

On-line diarization is accomplished by a two stage process, first, a fully unsupervised iterative diarization algorithm is applied over some Initial Training Set (ITS) of the audio stream, in this stage speakers and non-speech models are trained. On the second stage of diarization, trained models are used in order to perform on-line segmentation and labeling of the audio-stream.

A block diagram of the baseline diarization system used during the first stage of the diarization process is given in Figure 1.

Assume an ITS of N samples from the audio stream $s^N(n)$. The initial training set is first pre-processed using standard pre-emphasis filter, $f(z) = 1 - 0.95z^{-1}$. Mel Frequency Cepstral Coefficients (MFCC) features are then extracted from the ITS using 20mSec frames with 10mSec overlap between consequent frames. Twelve MFCC features are extracted (excluding c0) from each frame. The contribution of delta-features to the overall diarization was also investigated, thus, experimentations were conducted twice.

2.1. First-Stage Initialization

As there is no prior information regarding either one of the speakers nor the environmental conditions and non-speech, an initial assignment algorithm is required in order to initialize speakers and non-speech models to be used in the unsupervised diarization process. An initialization algorithm is suggested in [10], namely Weighted Segmental K-Means initialization (WSKM). Weighted Segmental K-Means is basically composed of a thresholded energy detector for speech/non-speech classification followed by a variant of the K-Means clustering algorithm used to cluster segments labeled as speech. Non-speech segments are then clustered to construct the non-speech initial cluster, speakers initial clusters are constructed in accordance with the initialization algorithm.

Weighted segmental K-Means algorithm is described in Al-

gorithm 1. Algorithm 1 only states the outline for applying WSKM, for a full description of the algorithm and relative performance comparison to other initialization algorithms, see [10].

Algorithm 1 Weighted Segmental K-Means initial Assignment (WSKMA)

Require: **ITS** $= s^N(n)$, initial training set samples. $\mathbf{O} = \{O_k\}_{k=1}^{K}$, a set of features extracted from the initial training set **ITS**

1: Perform an initial speech/non-speech segmentation.
2: Mark non-speech segments by $\{NS_j^{l_j}\}_{j=1}^{J}$ and speech segments by $\{S_i^{l_i}\}_{i=1}^{I}$ where l_j and l_i are the lengths of the segments such that $\sum_{j=1}^{J} l_j + \sum_{i=1}^{I} l_i = N$.
3: Estimate the mean for each speech segment $\{SC_i\}_{i=1}^{I}$, where SC_i is the estimated mean of the i^{th} speech segment.
4: Assign a weight $w_i = l_i$ to each of the means $\{SC_i\}_{i=1}^{I}$.
5: Initialize K-Means centroids, $\{V_r\}_{r=1}^{R}$
6: Estimate the new centroids using K-Means algorithms such that $V_r^{new} = \frac{\sum_{SC_i \in Cluster_r} w_i SC_i}{\sum_{SC_i \in Cluster_r} w_i}$
7: For all $\{SC_i \in Cluster_r\}_{i=1,...I,r=1,...R}$ assign $\{S_i \in Cluster_r\}_{i=1,...I,r=1,...R}$.

Following the application of the WSKM algorithm, speakers and non-speech initial clusters are generated.
Speakers and non-speech initial clusters are then trained using a Self Organizing Map (SOM) [12] of 6×10 neurons followed by an iterative adaptation and re-segmentation process.

2.2. SOM Based Vector Quantization

Speakers and non-speech models in this study are based on a non-parametric Self Organizing Map (SOM) [12]. Although speakers in the literature are almost always modeled using a statistical model, e.g. a mixture of statistical kernel functions, which is generally a Gaussian Mixture Model (GMM) [13], for short segments, there might not be sufficient statistical data to train either the speakers or non-speech models. Light weight and more efficient model is achieved using SOM models, while preserving diarization accuracy. Using SOM models, each speaker is modeled by a Code Book (CB) where each neuron in the CB is a Code Word (CW).

Given a set of feature vectors (observations) $\mathbf{O} = \{O_k\}_{k=1}^{K} \in \mathbb{R}^d$, an iterative algorithm for SOM training is presented in Algorithm 2.

Once speaker and non-speech model are generated, a distance or distortion measure is required in order to perform some time-series clustering of the data. Distortion measure is achieved through VQ as a likelihood estimator [14].

Having R Code Books (CB) and C Code Words (CW) in each CB, log-likelihood of the data can be estimated under the following assumption: for each CB, $\{CB_r\}_{r=1}^{R}$ each CW, $\{CW_l\}_{l=1}^{L}$ is the mean of a Gaussian probability density function (*pdf*) with a unit covariance matrix. Thus, log-likelihood of one observation can be estimated as follows: be a feature vector $o_k = [o_k^1, o_k^2, ..., o_k^d]^T \in \mathbb{R}^d$, where T is the transpose operator, and $CW^l = [cw^{l,1}, cw^{l,2}, ..., cw^{l,d}]^T \in \mathbb{R}^d$, then:

Algorithm 2 Self Organizing Map Training

1: Initialization
- Set the size of the CB $\rightarrow C$
- Initialize reference vectors $\mathbf{v^0} = \{v_c^0\}_{c=1}^{C}$
- Set small and positive learning coefficients α and γ, and the "winner" neuron neighborhood E^j.
- Set the number of SOM training iterations J

2: **for** $j = 1, ..., J$ **do**
3: Randomly choose an observation $O(k_r)$
4: Find the "winner" (closest) neuron

$$v_{c^*}^j = \min_c ||O(k_r) - v_c^j||^2 \quad \forall c = 1, ..., C$$

5: Update the "winner" neuron $v_{c^*}^j$ and it's neighbor neurones $E_{c^*}^j$:

$$v_c^{j+1} = v_c^j + \alpha^j \left[O(k_r) - v_c^j\right] \quad i \in E_{c^*}^j$$
$$v_c^{j+1} = v_c^j \quad i \notin E_{c^*}^k$$

6: Decrease the learning coefficient $\alpha^{j+1} = \alpha^j - \epsilon$
7: Decrease the neighborhood radius $E^{j+1} = E^j - \gamma$
8: **end for**

$$L(O_k|CB_r) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}(O_k - CW_r^{l^*,n})^T(O_k - CW_r^{l^*,n}) \quad (1)$$

Where

$$l^* = \arg \max_{l=1,...,L} \{(O_k - CW_r^{l,n})^T(O_k - CW_r^{l,n}) \quad (2)$$

Joint likelihood of all of the observations given CB_r is then:

$$L(\mathbf{O}|CB_r) = -\frac{dK}{2}\log(2\pi) - \\ -\sum_{k=1}^{K}(O_k - CW_r^{l^*,n})^T(O_k - CW_r^{l^*,n}) \quad (3)$$

2.3. HMM Time-Series Clustering

Time series clustering in this study is conducted using a modified Hidden Markov Model (HMM). Hidden Markov Model is a statistical finite-state machine characterized entirely by model parameters $\lambda = (A, B, \pi)$, where A is the state transition probabilities matrix, B states the observation likelihood (emission) matrix and π states the initial probabilities for each state, HMM's with applications in speech recognition are extensively reviewed in [15]. Although speech recognition and speech diarization share some common properties, HMM as described in [15] could not be applied without some modifications.

Modifications to the HMM are in the physical restriction over speaker turns. Though finely tuned HMM could handle time-series clustering of the conversation with fair accuracy, diarization is generally performed in absence of such tuned model, HMM parameters then must be estimated from the data. In order to provide some constraints for parameter estimation, a minimum duration τ is enforced over speaker turns, it is assumed that once speaker r has commenced speaking, he/she would continue speaking for at least τ seconds.

Such HMM can be described using a hyper-state transition matrix. Assume each speaker and non-speech forms a hyper-state in a HMM as shown in Figure. 2.

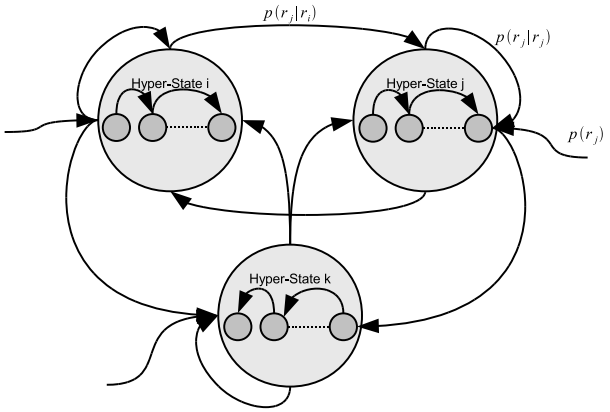

Figure 2: *Hyper state HMM.*

Hyper-state transition matrix A is a block matrix:

$$A = \begin{pmatrix} a_{r_1,r_1} & a_{r_1,r_2} & \cdots & a_{r_1,r_R} \\ a_{r_2,r_1} & a_{r_2,r_2} & \cdots & a_{r_2,r_R} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r_R,r_1} & a_{r_R,r_2} & \cdots & a_{r_R,r_R} \end{pmatrix}_{R\tau \times R\tau} \quad (4)$$

With diagonal elements which are hyper-state transition matrices:

$$a_{r_i,r_i} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p(r_i|r_i) & 0 & 0 & \cdots & 0 \end{pmatrix}_{\tau \times \tau} \quad (5)$$

and off-diagonal elements, which are inter-hyper-state transition matrices:

$$a_{r_i,r_j} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p(r_i|r_j) & 0 & 0 & \cdots & 0 \end{pmatrix}_{\tau \times \tau} \quad (6)$$

By adhering to this approach, standard time-series clustering accomplished by the Viterbi algorithm [15] can be used for speaker diarization. Construction of the observation matrix is through the use of SOM as a likelihood estimator discussed in the previous sub-section. Initial probabilities are uniformly set as $\frac{1}{R}$. Hidden Markov Model parameters are updated on each consecutive iteration of the diarization system, thus, approaching the desired tuned HMM which will be used on the second stage of on-line diarization.

Speaker models are trained on each iteration of the diarization system in accordance with the Viterbi path. Using this approach speaker models are also adapted as to better describe speakers and non-speech properties.

2.4. Unsupervised Diarization

Diarization of the ITS is accomplished by following Algorithm 3.

Algorithm 3 Unsupervised Diarization

1: Initialization
- Set the number of iterations $\rightarrow I$
- Set the minimum duration constraint $\rightarrow \tau$
- Set speaker and non-speech initial clusters $\{S_r\}_{r=1}^{R+1}$

Train $\mathbf{M} = \{M_r\}_{r=1}^{R+1}$ models from the initial cluster using the SOM algorithm
- Initialize HMM

2: **for** $j = 1, ..., I$ **do**
3: Apply the Viterbi algorithm using $R+1$ models and the set of observations observations $\mathbf{O} = \{O_k\}_{k=1}^{K}$
4: Cluster the conversation into $R+1$ clusters in accordance with the Viterbi path
5: Train $\mathbf{M} = \{M_r\}_{r=1}^{R+1}$ models from previous stage clusters using the SOM algorithm
6: Update HMM parameters in accordance with the Viterbi path
7: **end for**

Once ITS is fully processed, $\mathbf{M} = \{M_r\}_{r=1}^{R+1}$ speaker and non speech models as well as a tuned HMM are available, these will be used for the rapid on-line diarization stage. Both speakers and non speech model and HMM parameters would certainly be more accurately trained to fit source properties given the entire body of data, however, lacking all of the data, using part of the data to update model parameters seems a reasonable enough approach.

3. On-Line Diarization

On-line diarization in this study is accomplished by applying unsupervised diarization over an ITS of the audio stream (this segment length states the principle delay of the on-line diarization system), followed by segmentation and indexing of the entire audio stream using the Viterbi algorithm.

A block diagram of the on-line diarization system is given in Figure 3.

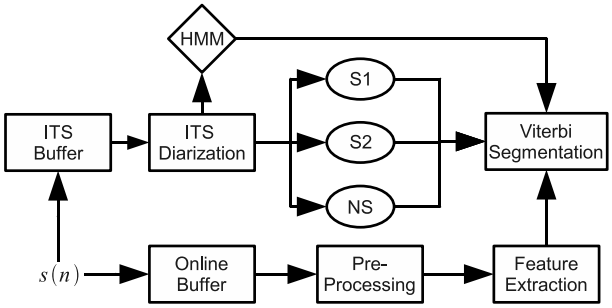

Figure 3: *On-line diarization system.*

Where $S1, S2$ and NS are speaker 1, speaker 2 and non-speech models respectively. HMM states the tuned Hidden Markov Model parameters following ITS diarization. Conversation samples are applied as input to both ITS diarization and to the online segmentation such that pre-processing and feature extraction could be performed in real time. Note that given the speakers and non-speech models along with HMM parameters, the on-line diarization system is entirely independent of the initial unsupervised diarization system.

Feature extraction and pre-processing are always performed

in an on-line manner. Once the unsupervised stage of the diarization has concluded, calculation of the emission probability matrix and the forward stage of the Viterbi algorithm could also be performed on-line. Thus, in order to provide diarization results, only backtracking in the Viterbi algorithm has to be performed, this can be accomplished in a few milliseconds.

3.1. Complexity Consideration

On-line diarization relies heavily on the rapid backtracking stage of the Viterbi algorithm. Pre-processing and feature extraction is always performed in an on-line manner, and once the processing of ITS has concluded, calculation of the emission probability matrix as well as the forward stage of the Viterbi algorithm could be performed on-line as well.
This way, once required, the segmentation and labeling of the audio stream until a required time t_r could be accomplished in several milliseconds.

Backtracking in the Viterbi algorithm is basically composed of two memory read and one memory write operations (to the Viterbi path). Time requirements for the Viterbi algorithm as a function of conversation length is given in Figure 4, note that fluctuations arise due to the use of a non real-time linux operating system and the short duration of the backtracking stage.

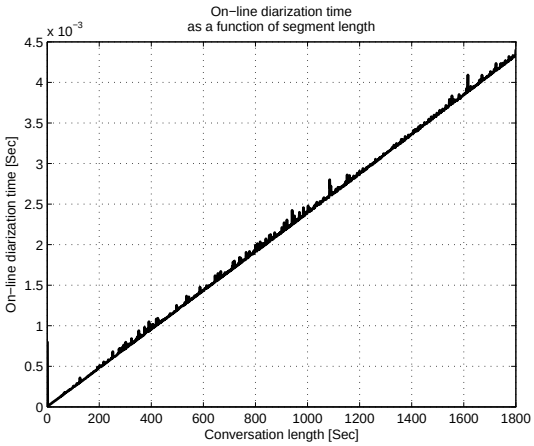

Figure 4: *On-line diarization time.*

The backtracking stage is shown to exhibit time-linear properties and even for a very long 30Min conversation, the backtracking stage requires less than 4.5mSec. This is less than the window length required for feature extraction (10mSec) so, on-line diarization could also be applied in real-time. For short conversations (under 60 Sec) the first stage of diarization takes 2Sec to accomplish. Thus, while processing very short conversations, no on-line diarization stage is required.

4. Experimentations and Results

The suggested on-line diarization was applied over 2048, 5Min length recordings extracted from the NIST 2005 Speaker Recognition Evaluation (SRE) [11]. Recordings are of two speaker conversations recorded using a two microphone channel (4-Wire) at a sampling frequency of 8kHz, the channels are summed and normalized in order to generate a single channel audio stream (2-Wire).

Twelfth order MFCC features are extracted from each audio stream (excluding c0), including and excluding delta-features, thus, the experiment was conducted twice.

The entire database was first processed by the unsupervised diarization stage in order to generate a lower bound for the on-line diarization results. On-line diarization was then applied to the entire database using variable length ITS.

4.1. Diarization Error Measurement

Diarization error is generally measured using the Diarization Error Rate measure as defined by the NIST Rich Transcription evaluation [16]. Diarization error rate measures the fraction of the time not attributed correctly to either one of the speakers or non-speech.

Assume segments in the segmented conversation $\mathbf{C} = \{C_s\}_{s=1}^{S}$, DER is measured using equation 7

$$DER = \frac{\sum_{s=1}^{S} dur(C_s) \cdot (max(N_r(C_s), N_h(C_s)) - N_c(s))}{\sum_{s=1}^{S} dur(C_s) \cdot N_r} \quad (7)$$

Where:

- $N_r(C_s)$ states the number of speakers in segment C_s stated by the reference diarization
- $N_h(C_s)$ states the number of speakers in segment C_s stated by the hypothesized diarization
- $N_c(C_s)$ states the number of speakers in segment C_s that were correctly assigned by the diarization system.

For telephone conversations diarization, only two speakers exist, however, two speakers conversing at once, that is, overlapped speech, must also be taken into account. The suggested diarization system does not currently handle overlapped speech. Segments labeled as overlapped speech by the reference diarization are always in an error state (current diarization system assigns segments to one of two speakers or to non-speech). That is, the error incurred by overlapped speech is added to the overall DER. Moreover, non-speech is also taken as one of the models while evaluating DER.

4.2. Results

Diarization error rate as a function of ITS length evaluated with and without delta-features are given in Figure 5.

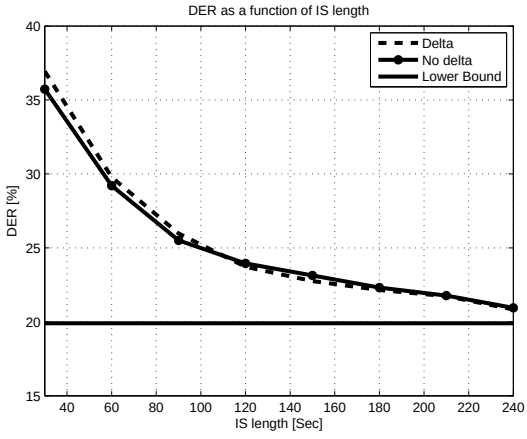

Figure 5: *Online diarization results.*

As previously suggested, it can be seen that delta-features does not contribute to lower DER. For short ITS, using delta-features is shown to increase DER.

On-line diarization seems to approach the suggested lower bound as the ITS length is increased, e.g., using 120 Sec ITS length provides 23.9% DER, while using 180 Sec ITS length provides 22.3% DER.

5. Conclusion

On-line diarization is implemented through a two-stage diarization algorithm. In the first stage, unsupervised, iterative diarization is applied over some initial training set extracted from the conversation in order to produce speakers and non-speech models as well as tuned HMM parameters. The second stage of diarization consists of time-series clustering via the Viterbi algorithm. This stage employs the models and HMM generated using the first stage of diarization to rapidly segment the conversation when required or when the conversation ends.

Short conversations (shorter than 30Sec) should only be processed by the first stage of diarization, providing diarization results 2Sec following conversation conclusion.

Applying the diarization system over 2048 conversations extracted from the NIST 2005 speaker recognition evaluation while using 180Sec ITS length followed by on-line diarization to the entire conversation provided 22.3% DER. this is roughly 2.4% higher than the lower bound attained by applying first stage (unsupervised) diarization over the entire conversation.
The diarization system suggested does not require any a-priori given information regarding the speakers or the environmental conditions/channel. No other parameter is required to be set prior to the application of the diarization system, these properties makes the suggested diarization system highly robust and scalable.
Further improvements could be applied to the suggested diarization system by handling overlapped speech detection and by adaptation of speakers and non-speech models as the conversation continues.

6. References

[1] S. Chen and P. S. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, USA, February 1998.

[2] V. Gupta, P. Kenny, P. Ouellet, G. Boulianne, and P. Dumouchel, "Combining Gaussianized/Non-Gaussianized Features to Improve Speaker Diarization of Telephone Conversations," *Signal Processing Letters, IEEE*, vol. 14, no. 12, pp. 1040–1043, November 2007.

[3] C. Costin and M. Costin, "New attempts in sound diarization," in *Soft Computing Applications, 2009. SOFA '09. 3rd International Workshop on*, September 2009, pp. 71–76.

[4] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1557–1565, August 2006.

[5] K. Markov and S. Nakamura, "Never-ending learning system for on-line speaker diarization," in *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, December 2007, pp. 699–704.

[6] D. Lilt and F. Kubala, "Online speaker clustering," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, May 2004, vol. 1, pp. I–333–6 vol.1.

[7] T. Koshinaka, K. Nagatomo, and K. Shinoda, "Online speaker clustering using incremental learning of an ergodic hidden Markov model," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, April 2009, pp. 4093–4096.

[8] N. Dhananjaya and B. Yegnanarayana, "Speaker change detection in casual conversations using excitation source features," *Speech Communication*, vol. 50, no. 2, pp. 153–161, 2008.

[9] R. Dehak, N. Dehak, P. Kenny, and P. Dumouchel, "Linear and non linear kernel GMM supervector machines for speaker verification," in *Proc. Interspeech*, 2007, number 3, pp. 302–305.

[10] O. Ben-Harush, I. Lapidot, and H. Guterman, "Weighted Segmental K-Means Initialization for SOM-Based Speaker Clustering," in *INTERSPEECH 2008*, 2008.

[11] "NIST Speaker Recognition Evaluation, http://www.itl.nist.gov/iad/mig/tests/sre/," .

[12] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, August 2002.

[13] A.P. Dempster, N.M. Laird, D.B. Rubin, and Others, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 138, 1977.

[14] I. Lapidot, "SOM as likelihood estimator for speaker clustering," in *EUROSPEECH 2003*, 2003.

[15] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, August 2002.

[16] "NIST Rich Transcription evaluation, website: http://www.nist.gov/speech/tests/rt," .