New Attempts in Sound Diarization

Ciprian Costin*, Mihaela Costin**
* “Al. I. Cuza” University, Department of Computer Science, Iasi, Romania
** Institute of Computer Science, Romanian Academy, Iasi Branch
E-mail: ciprian.costin@info.uaic.ro, mccostin@iiti.tuiasi.ro

Abstract - The paper discusses a new hybrid method in sound diarization (the process of segmenting an audio file into chunks that represent unique sources and clustering the obtained segments into groups that represent the same item). The most recent results are focusing mainly on the identification of voices during the telephonic recordings. In the hybrid method proposed here, a clustering is applied first, using an agglomerative approach regarding the construction of speaker models. Subsequently, when consistent amounts of data are gathered, special models are built using speaker factors. This idea gives good performance over the classical approach as the low-level clustering Bayesian Information Criterion scheme has poor performance on complex models, where speaker factors have very good precision. Speaker diarization improves speaker verification for multi-speaker audio (summed channel telephone data, single microphone interview data), is very important for speech recognition, and improves readability of an automatic transcription by structuring the audio stream into speaker turns and in some cases by providing the identity of the speakers. Sound diarization offers information which can be of interest for the multimedia documents indexing, in human-computer interaction, robotics, security systems, etc.

I. INTRODUCTION

Audio/speaker diarization might be defined in various ways, depending on the degree of process complexity, but a common core of the definitions remains, referring to the succession of different voices, sounds or noise and their identification. For example, a comprehensive definition is related to the energy carried by the sound waveform and the time intervals:

“Audio diarization is the process of annotating an input audio channel with information that attributes (possibly overlapping) temporal regions of signal energy to their specific sources. These sources can include particular speakers, music, background noise sources, and other signal source/channel characteristics.” [1] When related to voice, in the literature, this is a task also referred to as “who spoke when” and as speaker segmentation and clustering [1]: the process automatically marks where speaker changes occur in speech and associates segments of speech coming from the same speaker.

In a different formulation, from a programmer’s viewpoint, referring to voices, “speaker diarization is the process of segmenting an audio file into chunks that represent unique speakers, followed by clustering of the obtained segments into groups that represent the voices of specific persons, therefore input audio streams being partitioned into homogeneous segments according to the speaker identity” [2].

This process might increase the readability of an automatic transcription structuring the audio stream into speakers/sources turns [3], the technique being interesting for multimedia documents indexing, in security, in audio-video survey, in teaching or in medical, biological, behavioral research, in music, and so on.

While during a telephonic conversation, usually two voices are encountered (with the afferent possible background noise), in a quotidian situation more voices and superposed speech or sounds could appear. Major attempts are made in the direction of discovering the best way to analyze and display the exact sequence of voices during conversations (radio shows, telephonic or party speech), every field presenting its own important particularities. Having the exact delimitation of voices implies better further voice/speaker identification or verification, in the usual survey systems.

This work derives from the recent approaches developed during the authors’ participation to the Johns Hopkins Natural Language Processing Workshop, June-July 2008 [2].

Worldwide, researchers are developing simultaneously, parallel, methods which are presented during the workshops aiming to a more increased degree of standardization, as those organized by the National Institute of Standards and Technology (NIST), where the tasks are imposed and all the teams use the same databases for training and test purposes. Among the main research teams are MIT-LL - Massachusetts Institute of Technology - Lincoln Laboratory, CUED - Cambridge University Engineering Department, LIMSI - Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur, etc. The NIST Speaker Recognition Evaluation (SRE) series has the goal to contribute to the research and the calibration of the technical approaches for text independent speaker recognition, measuring the state-of-the-art technology and finding the most promising algorithms to drive the research forward [3]. There is a periodic evaluation every two years, NIST reporting the official results following each workshop, according to the rules and tasks previously established [3].

II. RECENT RESULTS ON SPEAKER DIARIZATION

In a short overview, we have to underline the fact that the audio diarization task is only an issue in the challenging and intensively debated problem of real time speech and speaker recognition. In speaker detection systems, the inter-channel and inter-session variability have a great influence on the final results. The type of microphones, the transmission channel, the modifications inevitably appearing in speaker voice (health state, aging and mood) and the acoustic background might influence very much the results. Using the
same microphone in training and in test, it was stated that the error is smaller (ERR<1%) than in the situation of using different microphones in training and test (ERR<3%) [4]. Inter- and intra-speaker acoustic differences are also important. Inter-speaker variability is determined by the age, sex, nationality (depending on the native language there are different manners of pronouncing vowels and consonants), but mainly by the physical factors (the form and the volume of the five resonant cavities of the vocal tract are determining the quality of the vocal sound). Intra-speaker differences refer to different utterances of the same word according to the emotional or physical state, or due to the necessity of emphasizing a word or an idea. Acoustic difference may also occur because of the channel variability, consisting in effects of the environmental noise (constant or transient) and of the transmission channel noise (data channels, microphones or telephone lines).

For understanding the necessity of pushing forward the research in this domain it is interesting to list the previous results claimed in the Spring 2007 (RT-07) Rich Transcription Meeting Recognition Evaluation from the series of community wide evaluations of language technologies in the meeting domain (the most recent meeting was in 2009 RT Meeting Recognition Evaluation, May 28 - 29, Florida Institute of Technology, with results not yet available) [3]. The SASTT task (speaker attributed speech to text), a new evaluation task imposed for the 2007 session, is combining STT (speech to text transcription) and SPKR tasks (speaker diarization). The test data consisted of three test sets: conference meetings, lecture meetings and breaks from lecture meetings (condition imposed as a test set in 2007 competition) [3]. For SPKR (the “Who Spoke When” Diarization task), the lowest diarization error rates for all speech in the multiple distant microphone condition were: 8.5% for the conference, 25.8 % for the lecture, and 25.5 % for the coffee break test sets, for up to four simultaneous speakers in the multiple distant microphone condition [5]. In SPKR systems clustering task, it is necessary to annotate a meeting with intervals of time indicating when a participant is speaking and to cluster those regions [5].

The paper presents some contributions in implementing a new hybrid method for speaker diarization, derived from the classical approach. The first practical trials were made on NIST Summed Channel telephone data, during the Johns Hopkins Natural Language Processing Workshop, June-July 2008. In the first part of the paper we present a short overview regarding the main attempts in speaker diarization. This is followed by comments on the results obtained during the participation to JH University, NLPW using a new hybrid approach.

III. STATE-OF-THE-ART TECHNIQUE IN SPEAKER DIARIZATION

A new hybrid method for speaker diarization, based on the classical approach (segmentation, clustering and Viterbi Resegmentation) is discussed; therefore it is important to review the state-of-the-art technique in speaker diarization up to the 2008 Workshop in Natural Language Processing at the Johns Hopkins University, which included the following steps:

1. Creating feature files from raw (unprocessed) audio
2. Applying speaker segmentation (speaker change detection)
3. Applying speaker clustering on the output of 2.
5. Reiterating steps 2-4 until the system becomes stable.

Feature files are created from raw audio, using HTK tools. We take 100 frames/second with 10 ms overlapping (therefore frame length is 20ms). For speaker diarization, MFCC values carry important data concerning speaker identity. While it is demonstrated that considering delta and double-delta MFCC produces a performance benefit in speaker verification [6], we use only MFCC features, without C0, delta and double-deltas, as the algorithm used does not get any improvement from those derivatives. It is relevant to use the NIST ASR (Automatic Speech Recognition) System for SAD (Speech Activity Detection), because it is not efficient to analyze silence zones. We have to underline that the silence zones reduce the informational density of the signal analyzed: for a specified signal length: the more of it is silence, the lower the information contained in the signal is. Hence, silence elimination is important in avoiding the analysis of meaningless signal zones. This improves the efficiency of the processing algorithms, especially when the amount of data is important and the ratio between silence length and total signal length is significant.

For speaker segmentation, we take into account an initial window size of $z$ frames. The system takes any hypothesized change point $i$ from the window and tests the penalized GLR (Generalized Likelihood Ratio):

$$\Delta BIC(i) = -\log \frac{P(z \mid \lambda_\alpha)}{P(x_i \mid \lambda_\alpha)P(y_i \mid \lambda_\alpha)} - \alpha P .$$  

$BIC$ stands for the Bayesian Information Criterion, $\lambda$, represents the probability density function model for segment $z$, and the penalty $P$ (where $N$ is the number of frames in the window and $d$ stands for full Gaussian Dimension) is used because the denominator has twice as many parameters as the numerator. $P = 1/2(d+1/2 \log (d+1)) \log N$ , (more details can be found in [7], [8]). Basically, we test if it is better to keep the whole segment as one speaker or to divide it in two parts, each one representing a certain speaker. If a $\Delta BIC > 0$ is found within the search window, we have a speaker change point. The process is restarted from that point. If not, the window is gradually increased until it reaches its maximal size (usually 5 seconds). If a speaker change point is not found within the maximal window, we skip it, considering it as belonging to one speaker and apply the algorithm for the next window (this step generates a file with around 300 lines for a 5-minute conversation, each line having two values, representing the time when the person started and respectively stopped speaking).
We could have considered only 1 second chunks and gone directly to do speaker clustering, but experiments found that errors made in the segmentation step are not only difficult to correct later, but also degrade the performance of the subsequent clustering step [8], [9]. So it is important to have good results on the initial speaker segmentation.

For speaker clustering, the system uses an agglomerative clustering algorithm, as generally bottom-up systems perform better over top-down versions (it does not make much sense to divide at the first step a 15 minute segment into two parts, each one representing a different speaker). At each stage we analyze the distance between all pairs of clusters and take the minimal $\Delta BIC$ (we use the same $BIC$ used in the speaker change detection system). Small $\Delta BIC$ means very similar chunks of data, while a positive, big $\Delta BIC$ means very different voices. The algorithm stops when minimal $\Delta BIC$ reaches a value above 0 (zero), but we can force it to stop at a certain number of clusters (This step adds to the previously generated file a new data field for each line, the speaker label, which has the range from 0 to $n-1$, $n$ being the number of speakers).

For continuing the description of the next part of the algorithm, it is useful to remind that a GMM (Gaussian Mixture Model – [10]) is used in speaker recognition applications as a generic probabilistic model for multivariate densities capable of representing arbitrary densities, which makes it well suited for unconstrained text-independent applications. In the next phase, the Viterbi dynamic programming algorithm is used for finding the most likely sequence of hidden states (Viterbi path) that results in the sequence of observed events [6].

Consequently, for the Viterbi resegmentation, from the output of the clustering GMM models [10]-[14] are trained for each of the two clusters using an EM (Expectation Maximization) algorithm, iterated 10 times (note that by not considering an UBM (Universal Background Model - all the points in the speaker hyper-dimensional space) the system becomes more portable). For each frame ($t_i$), we consider $\log P(x_t | \lambda_1)$ and $\log P(x_t | \lambda_2)$, where $x_t$ is a specific frame and $\lambda_1$ and $\lambda_2$ are the speaker models built up to that point. We apply a dynamic programming using no cost for remaining in the same state and a high cost penalty for transitioning from one state to the other (we do not want the system to fluctuate very often – at a frame level, which is practically impossible in human conversation, yet we do not want to be rigid in the decision).

Then we retrain GMMs using frame assignment from most likely path, and iterate the whole algorithm of the subsystem a finite, predefined number of times.

For good performance, the algorithm needs to be iterated a considerable number of times (we considered 20 iterations). After this amount the system is very stable. However, there are some cases when performance degrades after more than 20 iterations, but those cases are quite limited in number. Details on the Viterbi Resegmentation performance on different number of iterations can be found in table 2.

For scoring the performance of the system, the Diarization Error Rate (DER) is the primary metric. DER is the ratio of incorrectly assigned speech time, (falsely detected speech, missed detections of speech, and incorrectly clustered speech) to the total amount of speech time, expressed as a percentage. A score of zero indicates perfect performance and higher scores indicate poorer performance.

<table>
<thead>
<tr>
<th>Number of Viterbi Iterations</th>
<th>Number of Audio Files</th>
<th>Diarization Error Rate (Average)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2213</td>
<td>19.85%</td>
<td>13.47%</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>8.91%</td>
<td>13.86%</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>7.50%</td>
<td>12.94%</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>7.06%</td>
<td>12.52%</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>6.90%</td>
<td>12.36%</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>6.82%</td>
<td>12.28%</td>
</tr>
</tbody>
</table>
The system diarization file is compared with the ground truth reference file using the following easy formula:

\[
\sum_s durs(s) \cdot (H_{\text{miss}}(s) + H_{\text{fa}}(s) + H_{\text{spe}}(s)) \bigg/ \sum_s durs(s) \cdot H_{\text{ref}}(s)
\]

where \(s\) is the longest continuous piece of audio for which the reference and hypothesized speakers do not change, \(durs(s) = \text{duration of } s\), \(H_{\text{miss}} = 0\) except \(H_{\text{miss}} = 1\) for missed, false alarm or speaker error segments, \(H_{\text{ref}} = 1\) for a segment containing a reference speaker. Incorrectly clustered speech, a speaker error, occurs when a system successfully detects speech but assigns the speech to the wrong speaker (see Figure 2).

**Figure 2.** Possible errors in a speaker diarization system (simple case - two voices)

The correct speaker attribution is detected by a minimal cost, in a one-to-one mapping from the speaker clusters to reference speaker clusters [15], [4]. A diagram of the baseline diarization system can be viewed in Figure 3.

**Figure 3.** Block diagram of overall diarization system

The configuration for the segmentation is: initial search window of 1 second, expandable up to 5 seconds by using a recursive function (for the experiments we constantly increase it with 0.5 seconds at a time), the small speech segments below 1 second being skipped (because there is a small chance of having a speaker change in such a short delay), the BIC threshold is 0 and the BIC weight is 1.0.

In the configuration for the speaker clustering stage, the minimal segment length to consider is 0.5 seconds, the BIC weight is 1.0 and the system set to stop at 2 clusters.

In the configuration for the Viterbi resegmentation, the number of iterations is 20, the EM (Expectation Maximization) algorithm number of iterations is 10, the GMM size is 32, and the dynamic programming transition cost is -46 (Figure 1).

**IV. HYBRID CLUSTERING USING SPEAKER FACTORS**

In order to understand the theory behind this approach, Joint Factor Analysis (JFA) explained in [11], we begin by recapitulating the basic assumptions in factor analysis. As in the papers of Kenny, using the same method and notations, \(C\) is the number of components in a Universal Background Model (UBM) and \(F\) the dimension of the acoustic feature vectors. The supervector is a term which refers to the CF-dimensional vector resulted from the concatenation of the \(F\)-dimensional mean vectors in the GMM for a certain utterance [11], [16], [17]. Assume that a speaker - and channel - dependent supervector \(M\) could be:

\[ M = m + vy + dz + ux \]  

where \(m\) is a \(CF \times 1\) supervector; \(v\) is a rectangular matrix of low rank and \(y\) is a normally distributed random vector; \(d\) is a \(CF \times CF\) diagonal matrix and \(z\) is a normally distributed \(CF\)-dimensional random vector. The columns of \(v\) are referred as *eigenvoices* and the components of \(y\) are the *speaker factors*.

Assume that the distribution of \(c\) has a hidden variable description of the form \(c = ux\), where \(u\) is a rectangular matrix of low rank and \(x\) is a normally distributed random vector. We refer to the components of \(x\) as channel factors and we use the term eigen-channels to refer to the columns of \(u\). Experiments done at LIMSI-CNRS have shown that by stopping the agglomerative clustering at a certain stage and then training GMMs until reaching the desired number of clusters, better results are obtained in comparison with those generated by the classical approach.

This new approach is based on the following idea: using BIC stopping criterion is good for low-level data (small chunks of speech), while comparing large clusters with BIC is inappropriate. So it makes sense to compare detailed speaker models using more complex methods, considering the hypothesis that more sophisticated models will perform better for larger amounts of data (both the GMM-UBM and the JFA methods [11], [16]-[18], have good performance over bigger amounts of speech). In this approach, we do the classical, agglomerative clustering up to a certain level (we tested many levels, and the best results are where the level is between 10 and 20). Subsequently, when consistent amounts of data (for good speaker models) are gathered, those special models are build using speaker factors.
So, we compute those $Y$ vectors of length 300 (one for each speaker model) and combine two clusters that had the maximal similarity with respect to the cosine distance.

$$d(x, y) = \sqrt{\sum_{i=1}^{\text{size}(Y)} (x_i - y_i)^2}$$

Among the modalities considered for computing the distance between $y$ vectors, the cosine distance was chosen due to the extremely fast computation time, as well as the excellent performance in speaker verification (1.85% - 2% EER – Equal Error Rate). The algorithm is repeated until the desired number of clusters is attained (Figure 4). Another version of the same idea is based on the results obtained in the first hybrid approach. Due to the nature of speech signal, certain noises can be detected as separate speakers. The duration of such noises is quite short, but, as mentioned above, computing $y$ vectors from small chunks of data gives bad results. In quite a few cases, when analyzing a conversation between two speakers, we ended by getting three speaker models, one being a specific noise; at the last step, the algorithm combined the two real speakers, because the noise was very different from the speaker models. So, instead of stopping at a certain level, we did the agglomerative clustering algorithm up to the top level, then extracted, at different levels, the clusters that had at least a specific number of frames (100 was the chosen number) and applied the clustering scheme using the $y$ vectors until two speaker models are achieved. This way, we are surely not computing $y$ vectors on too small clusters.

Figure 4. Diagram of first version of hybrid diarization

The main steps of the algorithm are presented in the following sequences.

The first method of hybrid clustering is the Level Cutting Hybrid Diarization (LCHD):
- Step 1: from the output of the segmentation (each line in format start_time - stop_time) run classical clustering algorithm up to the top level (until 2 clusters are obtained)
- Step 2: from each cluster run subroutine to get $Y$ vectors (of dimension 300)
- Step 3: compute cosine distance for each pair of 2 clusters
- Step 4: select the 2 clusters that got maximal similarity with respect to the cosine distance (from step 3)
- Step 5: merge the two clusters
- Step 6: iterate steps 2-5 until 2 speakers are obtained.

The second method of hybrid clustering is the Tree Search Hybrid Diarization TSHD (variant of hybrid clustering):
- Step 1: from the output of the segmentation (each line in format start_time - stop_time) run classical clustering algorithm up to the top level (until 2 clusters are obtained)
- Step 2: Select from the tree those clusters with at least 100 frames (this would form the leaves of a new tree that will be built using speaker factors - $Y$ vectors)
- Step 3: for each cluster run subroutine to get $Y$ vectors (of dimension 300)
- Step 4: compute cosine distance for each pair of 2 clusters
- Step 5: select the 2 clusters that got maximal similarity with respect to the cosine distance (from step 4)
- Step 6: merge the two clusters
- Step 7: iterate steps 2-6 until 2 speakers are obtained.

Results of hybrid clustering applied on summed telephone speech might be observed in table 3.

**TABLE III**

<table>
<thead>
<tr>
<th>Hybrid Clustering Results on Summed Telephone Speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>System (2213 Audio Files)</td>
</tr>
<tr>
<td>Baseline Diarization System</td>
</tr>
<tr>
<td>Hybrid System 1 LCHD</td>
</tr>
<tr>
<td>Hybrid System 2 TSHD</td>
</tr>
</tbody>
</table>

**TABLE IV**

<table>
<thead>
<tr>
<th>Summed Telephone Speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average DER (Diarization Error Rate)</td>
</tr>
<tr>
<td>Perfect diarization</td>
</tr>
<tr>
<td>No diarization</td>
</tr>
<tr>
<td>Baseline</td>
</tr>
<tr>
<td>Baseline + soft cluster</td>
</tr>
<tr>
<td>Hybrid</td>
</tr>
<tr>
<td>Streaming</td>
</tr>
<tr>
<td>Variational Bayes (VB)</td>
</tr>
</tbody>
</table>

In table 4, it might be observed that speaker diarization
improves speaker verification for multi-speaker audio (5-6%). Baseline DER (Diarization Error Rate) reduced from 7.35% to 3.52% by new soft-clustering algorithm borrowed from Variational Bayes; Variational Bayes with 200 eigenvoices achieved 3.75% DER, while streaming system with just 20 eigenvoices achieved 4.54% DER. Due to the fact that the samples are from telephone conversations, every audio file contains data from two speakers. The software can analyze data from more than two speakers, but the NIST task concerned only telephone conversations between two individuals. The most relevant part of the table 3 is to realize the big importance of the use of diarization in speaker verification.

CONCLUSIONS

A large amount of trials has been done on a substantial number of files (2213) from summed channel telephone conversations provided by NIST [3]. Even if the tree search hybrid diarization system provided good results, the errors are about the same as those obtained by the baseline diarization system. The computational complexity makes us search better solutions, implying for the future trials a preferential use of the vocalic content of the voice. For certain types of audio files - interview data, more speakers - the baseline diarization system achieves 3.75% DER, while streaming system with just 20 features meant to complete the information content about the method and the actual results obtained in diarization, with a fusion of multiple systems. Trying to aggregate this performs better than the others, or trying a weighted decision, finding cases where one of the many diarization systems could try to build a cluster voting scheme [19]. This means that the baseline system uses a non-gaussianized feature set). Also, we could try to build a cluster voting scheme [19]. This means finding cases where one of the many diarization systems performs better than the others, or trying a weighted decision, as a fusion of multiple systems. Trying to aggregate this method and the actual results obtained in diarization, with features meant to complete the information content about the voice, taking into account only to the vocalic content (due to the particularities of the vocal tract which give the very high individual specificity) might lead to good results, according to our previous tests and publications [20]÷[22]. The diarization technique could also be applied in music to singing voices or to identify different instruments, with similar sonorities.

ACKNOWLEDGMENT

Special acknowledgments to the Speaker Recognition Team from Johns Hopkins University 2008 Workshop on Natural Language Processing, particularly to Mr. Douglas A. Reynolds, Senior Researcher, MIT - Lincoln Laboratory, who directed my work during the workshop, to Mr. Lukas Burget, Brno University of Technology, team leader, and to the team members, Mr. Patrick Kenny CRIM - Centre de Recherche Informatique Montreal Canada and Mr. Fabio Castaldo, Politecnico di Torino, Italy. Special thanks for the theoretical and practical issues that I managed to learn and to further apply in my work.

REFERENCES