Unsupervised Segmentation and Verification of Multi-Speaker Conversational Speech

Unsupervised Segmentation and Verification of Multi-Speaker Conversational Speech

Emanuele Dalmasso, Pietro Laface*, Daniele Colibro^, Claudio Vair^*

Politecnico di Torino, Torino, Italy*
Loquendo, Torino, Italy^

{Emanuele.Dalmasso, Pietro.Laface}@polito.it
{Daniele.Colibro, Claudio.Vair}@loquendo.com

Abstract

This paper presents our approach to unsupervised multi-speaker conversational speech segmentation.
Speech segmentation is obtained in two steps that employ different techniques. The first step performs a preliminary segmentation of the conversation analyzing fixed length slices, and assumes the presence in every slice of one or two speakers. The second step clusters the segments obtained by the previous step, estimates the number of speaker, and refines the segment boundaries using more accurate models.
We evaluated our algorithms on the speaker segmentation tasks proposed by the 2000 NIST speaker recognition evaluation where the proposed approach produces state-of-the art segmentation error rates and on the 2004 NIST multi-speaker conversation tests where we compare the verification performance using automatically segmented training data with the one obtained using single speaker data.

1. Introduction

Speaker segmentation is an important topic for speaker detection and tracking. Clustering conversational speech into sets of regions corresponding to a putative single speaker is a reasonable approach when the speaker turns are frequent and the duration of a turn can be very short, as happens in the CallHome Corpus collected by the Linguistic Data Consortium that has been used in the 2000 NIST speaker recognition evaluation [1].
Our approach to speech segmentation works in two steps that make use of different techniques, and does not rely on a-priori knowledge. The techniques are different because they account for the amount of data that can be used for obtaining reliable speaker model estimates. The first step does not look for speaker changes using a sliding variable length analysis window (as done for example in [2-3]). It performs, instead, a preliminary blind segmentation (as proposed for example in [4-6]). This blind segmentation of the conversation is carried out analyzing signal slices of fixed length.
The length of a slice is chosen so that it can be assumed that most of the times a maximum of two speakers are present in every slice. This assumption is reasonable because the second step, relying on more information, is able to update the number of the detected speaker segments and the preliminary segment boundaries in every slice. The second step clusters the segments obtained by the previous step, estimates the refines the segment boundaries using more accurate models. number of speaker participating in the conversation, and

We evaluated our algorithms on the speaker segmentation tasks proposed by the 2000 NIST speaker recognition evaluation, and on a 2004 NIST multi-speaker conversation test. Real-time constraints are ignored in this work, the offline segmentation process, however, is fast because it only takes on average about 6% of the duration of a conversation.
The paper is organized as follows: Section 2 describes the method for obtaining a preliminary segmentation from fixed length slices. The final speaker clustering, the estimation of the number of speakers and the refinement of speaker homogeneous intervals are illustrated in Section 3. In Section 4 we validate our approach presenting the experimental results. Conclusions and comments are given in Section 5.

2. Preliminary segmentation

We process the audio file to produce a feature vector of 23 Mel-cepstral coefficients every 10 ms according to [6]. Automatic end-point detection is performed to remove long chunks of silence. Silence or noise regions inside the endpointed conversations are eliminated using the same energy-based Voice Activity Detector that we use for robust speech recognition in noisy conditions. The remaining signal is processed in slices of fixed length.
For the conversational speech tasks we set the slice length to 60 seconds: our assumption is that in 60 seconds of conversation most of the times only the turns of one or two speakers are present, otherwise it would be difficult to reliably detect homogeneous segments of more than two speaker using a limited amount of data. Moreover, this assumption does not harm the possibility for the second step, relying on more information, to update the number of speaker segments in a slice and their boundaries.
Using the reference segmentation, including silences, of the first minute of all the CallHome multi-speaker conversations, we computed the number of slices where the contributions of all the K speakers included has a duration greater than T seconds. These statistics, summarized in Figure 1, confirm our assumption because there are no slices including 4 speakers where a speaker talks more than 5 seconds, and there are a few 3-speaker slices where a speaker contributes for more than 10 seconds. Moreover, in these statistics we did not take into account overlapping speakers segments, which occur more frequently when several speakers take part to the conversation. We point out that the assumption of a maximum of two

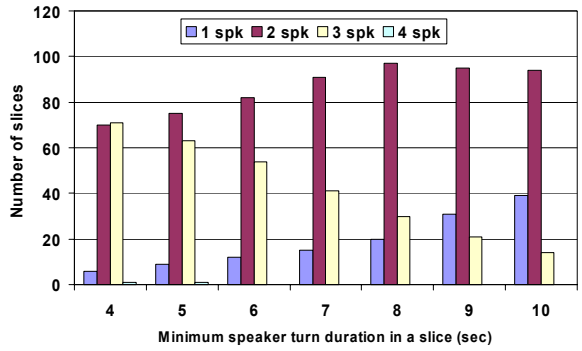

Figure 1: Number of speakers in the first minute of all the CallHome multi-speaker conversations versus the maximum duration of their turns.

speakers per slice is not critical because the structure of our preliminary segmentation is similar to the framework proposed in [5], where it is possible to automatically determine the "best" number of speaker per slice.

Every *slice* is partitioned into 60 analysis *windows* of one second that are processed using tied Gaussian Mixture Model density functions. The choice of tied GMM density functions allows very fast estimation, comparison, and adaptation of the speaker models [4,6].

2.1. World model

First we estimate, using all the frames in a slice, the parameters of the initial "world" GMM through vector quantization: we use 64 Gaussians for this model. This set of Gaussians is kept fixed for each window of the slice: only the mixture weights are re-estimated. Since the set of Gaussians is fixed, it is possible to compute once for all the emission probability of every Gaussian for each frame in the slice. The most significant emission probability contributions per frame only are stored, associated with the index of their corresponding Gaussian.

Few EM iterations are performed to re-estimate the initial weights. These operations are very fast because the number of processed frames is small, and because the number of emission probability contributions per frame, after pruning the insignificant ones, is on average 3.

2.2. Blind segmentation

A blind segmentation is performed using an ergodic 2 state HMM [4-5] with all the transitions costs are set to 0. The difference of our approach with respect to [4] is that, like in [5], we don't use neither supervised nor random initialization of the estimates of the HMM states. We initialize, instead, as suggested in [5] the state S_0 model by estimating the weights of the tied GMM model using the frames of a window, and the model of state S_1 using the frames of all the other windows of the slice. The weights are obtained, according to MAP estimation, as a linear combination of the weights estimated for the current window and the weights of the "world" model.

Few iterations of Viterbi decoding allow to re-estimate the two state models, and to label each window according to the best sequence of states.

Our approach for the preliminary segmentation differs from [5] because we use tied mixtures. This allows us to produce better initial speaker clusters, as will be described in the next sub-section.

Using tied mixtures, given the models λ_x and λ_y estimated with the frames of window x and y respectively, it easy to estimate the model λ_{xy} from the collection of the frames of both windows without performing any expensive likelihood computation. Since the free parameters to be estimated are the Gaussian weights only, they can be simply obtained according to [4] as:

$$g_k(w_{xy}) = \frac{1}{(n_x + n_y)}\left(n_x \cdot g_k(w_x) + n_y \cdot g_k(w_y)\right) \quad (1)$$

where w_{xy} refer to the segment that merges windows w_x and w_y including n_x and n_y frames respectively.

It is very fast, thus, to estimate first the model of state S_0 – i.e. the set of weights $\{g(S_0,w)\}$ – for every window w, and then to obtain the corresponding model for state S_1 interpolating the weights of the models according to (1). The same formula can be used during the process of re-estimation of the models of the two HMM states. MAP re-estimation is always performed due to the limited amount of data.

2.3. Selection of the best preliminary segmentation

In [5] the model of state S_0 is initialized with the parameters of the window that achieves the maximum likelihood given the "world" model. We have found that the choice of the seed model has significant influence on the segmentation: initializing the state S_0 model using the frames of different windows often produces quite different segmentations; this adversely affects the final speaker clustering as will be shown in Section 4.

Thus, an original contribution of this paper is a procedure to obtain the best preliminary segmentation.

Since the segmentation procedure is very fast, we estimate the initial models using, at each run, the frames of a different window. We obtain, thus, a set of different segmentations represented by strings of 60 labels, one per window. We want to generate a string, resulting from the combination of all these strings, which produces the maximum segmentation agreement among them. It is worth noting that the putative speaker label for a given window can differ in each segmentation string even if it represent the same speaker. For example, the following strings

1001101111…..
0110010000…..

represent the same segmentation because the two putative speaker labels (0 and 1) have been permuted.

Since different seed models will produce different segmentation strings, we must define a similarity measure between strings taking into account speaker re-labeling.

An optimal procedure for this task would be combinatorial with the number of the windows (60). We use, thus, a sub-optimal procedure, combinatorial with the number of speakers in a slice, which performs sequentially the optimization. This is possible because the segmentations differ, due to different initialization of the seed model, but their labels are not assigned randomly.

The procedure uses a matrix M with a row for each putative speaker s, and a column for each window w. $M(s,w)$ stores the

Table 1: Set *Z* of strings with permuted labels

1	**0**	**0**	**2**	**2**	**0**	**1**	**1**	**0**	**2**	**.**	**.**
0	**1**	**1**	**2**	**2**	**1**	**0**	**0**	**1**	**2**	**.**	**.**
1	**2**	**2**	**0**	**0**	**2**	**1**	**1**	**2**	**0**	**.**	**.**
2	**1**	**1**	**0**	**0**	**1**	**2**	**2**	**1**	**0**	**.**	**.**
1	**2**	**2**	**0**	**0**	**2**	**1**	**1**	**2**	**0**	**.**	**.**
2	**0**	**0**	**1**	**1**	**0**	**2**	**2**	**0**	**1**	**.**	**.**

number of times that window *w* has been assigned to speaker *s*.

The matrix is initialized as

$$M(z(w),w)=1 \qquad w=1,\ldots,60 \tag{2}$$

where *z(w)* is the first segmentation string.

Each segmentation string is processed as follows:

1) A set of $n=S!$ strings Z is generated accounting for all possible permutations of the speaker labels as shown in the Table 1 example for S=3 putative speakers.
2) For each string z_i in Z, a copy of the matrix M is created, and then updated according to

$$M_i(z_i(w),w)=M(z_i(w),w)+1 \qquad w=1,\ldots,60 \tag{3}$$

and total number of matching labels is computed as

$$matching_i=\sum_w \max_{s\varepsilon S}(M_i(s,w)) \qquad i=1,\ldots,n \tag{4}$$

Copying back the matrix b having best $matching_b$ value

$$M(s,w)=M_b(s,w) \tag{5}$$

the current best segmentation is obtained as

$$z(w)=\arg\max_s M(s,w) \tag{6}$$

This procedure finds, thus, the segmentation that maximizes the number of matching identifiers in each window for all the segmentations.

2.4. Speaker-homogeneous segments

The preliminary best segmentation is used to estimate the initial models of two speakers, then two iteration of Viterbi decoding are performed to produce the hypothesized speaker-homogeneous intervals, and to refine their boundaries. This procedure works frame-by-frame using a sliding window of 100 frames. It assigns to the central frame of the window the state identifier that maximizes the likelihood of the window frames given the state model. Again, this procedure is very fast because, for each window step, the decision requires only that the likelihood of a new frame is added and the likelihood of the oldest frame removed. A final step is performed to reassign short intervals (less than 10 frames) to the adjacent ones. The overall result is a set of intervals that identify two speaker-homogeneous segments.

2.5. Analysis of the speaker segments

Since it is possible that only one speaker has actually spoken in the current slice, the Bayesian Information Criterion is used to compare the likelihood of the two segments and to decide if they belong to different speakers, or if the entire slice has to be assigned to a single speaker.

The complete preliminary segmentation procedure is performed on every slice of the conversation.

3. Speaker clustering

The set of all the segments produced by the preliminary segmentation for every slices of the conversation is the input to the clustering algorithm.

3.1. Global world model

Since the amount of data in each putative speaker-homogeneous segment is not limited to one second, as was the case for the windows in the preliminary segmentation, it is possible to increase the resolution of the models to enhance their capability of clustering the segments.

Since we insist in using tied Gaussian mixtures, we must estimate a "global world" GMM.

To estimate the parameters of the initial "global world" GMM, we perform 3 iterations of a K-Means clustering algorithm using the frames of all the segments. The initial seeds for the K-Means algorithm are computed by clustering the mean vectors of the Gaussians belonging to the world models of all the conversation slices. The number of Gaussians for the enhanced model is set to 128 or 256 according to the total number of frames of the conversation. Again, only the significant emission probabilities for all the frames are stored.

A complete linkage hierarchical clustering is finally performed using a matrix of the distance between two segments whose element D_{IJ} is defined as

$$D_{IJ}=\frac{L_I+L_J-L_{IJ}}{\log(n_I+n_y)} \tag{7}$$

where $L_{x,}$ is the log-likelihood of the n_x frames of segments x given the model λ_x estimated on the same frames. L_{IJ} is to the log-likelihood of the union of the frames of segments I and of segment J given the model λ_{IJ} estimated as combination of the models λ_I and λ_I .

The agglomerative clustering merges similar segments until the distance between two clusters is greater than a fixed threshold depending on the number of Gaussians of the models: this correspond to a BIC similarity criterion [2]. The threshold value controls, thus, the number of speakers that are detected in the conversation and the speaker clusters. It is worth noting that, for the sake of efficiency, we do not re-estimate the cluster models and their likelihoods.

Finally, 2 iterations of Viterbi decoding are performed to refine the speaker models and the boundaries of the cluster of segments.

4. Experimental results

To assess our approach we addressed the speaker segmentation tasks of the NIST 2000 evaluation both on Switchboard and on the multilingual CallHome data [1].

All scores have been obtained using the scoring script *seg_scoring.v2.1.pl* provided by NIST, ignoring collar periods of 250 msec, as it is usual for these tests. The segmentation has been performed without knowledge of the number of speakers taking part to the conversations.

Table 2 compares the segmentation errors, on Switchboard data, using a single seed model for obtaining the preliminary segmentation of a slice, like in [5], rather the best preliminary segmentation illustrated in sub-section 2.4. These results show significant performance degradations due to an inaccurate initial segmentation. The same considerations can be made

considering the results of the same experiment performed on the CallHome evaluation data reported in Table 3. For these experiments we kept fixed the same thresholds used for the Switchboard. Our results on these tasks, are in line with to the ones reported in [1] for one of the best sites system participating to the NIST 2000 evaluation.

The performance improvement due to the detection of one or two speakers per slice (sub-section 2.5) can be compared with the results – shown in the last column of the table – that are obtained assuming that two speakers are always present in a slice.

The performance of our approach has also been assessed on the CallHome n-speaker segmentation subtask. Again our results, shown in Table 4, are comparable with the one presented in [1].

It can be noticed that, with the current setting of the threshold in the segment clustering algorithm, we slightly underestimate the actual number of speakers in conversations. Overall, the mean difference between the estimated number of speakers and the actual one is 0.36.

Finally, the speaker clustering procedure has been tested on a 2004 NIST multi-speaker conversation test, training speaker models with reference (3-sides) and automatically segmented (3-convs) data. Comparing the DET functions, shown in Figure 2, obtained testing the 1-side data with the two set of models, we can observe how the use of contaminated data affects our GMM speaker verification system. The two families of functions are produced by two GMM systems. The best one, however, is trained and tested using MFCC feature warping (FW) and world-model adapted to the NIST 2004 data. We observe that feature warping gives very good results, but that it has greater impact on the accuracy of impure speaker models, while models trained with un-warped features are more tolerant to segmentation errors.

Table 2– Segmentation errors on Switchboard data

Segment speakers	Best seed	All seeds
Male	18.0%	7.0%
Female	19.3%	11.4%
Mixed	14.6%	4.9%
Total	17.1%	7.6%

Table 3– Segmentation errors on CallHome data

N. of speakers	Best seed	All seeds	2 speakers per slice
2	17.1%	11.5%	13.4%
3	22.9%	18.5%	19.7%
4	20.2%	14.6%	16.5%
5	23.0%	18.7%	19.1%
6	33.2%	22.6%	23.4%
7	33.1%	28.4%	28.4%

Table 4– Number of speakers on CallHome data

Number of speakers	2	3	4	5	6	7
Average number of detected speakers	1.9	2.3	3.3	4.4	4.8	6.5

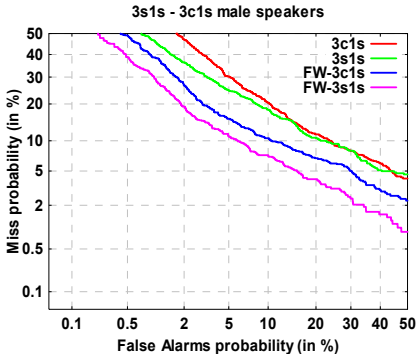

Figure 2: DET functions comparing contaminated and pure speaker models, and feature warped and un-warped systems

Conclusions[1]

An approach to unsupervised multi-speaker conversational speech segmentation has been presented that has shown its capability of detecting and segmenting conversations with unknown number of speakers. It tries to carefully exploit the available data performing a preliminary blind segmentation, analyzing conversation slices of fixed length. A second step, relying on more data, can use more accurate models to clusters the segments obtained by the previous step, to finds the number of speaker, and to refine the speaker segment boundaries. Good results have been reported for the multi-speaker segmentation and verification of conversational speech using standard data and tools.

5. References

[1] A. Martin, and M.A. Przybocki, "Speaker recognition in a Multi-Speaker Environment," *Proc. Eurospeech*, Vol. 2, pp. 787–790, 2001.

[2] S.S Chen, and P.S. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion," *Proc. Darpa News transcription and Understanding Workshop*, Vol. 6, pp. 127-132, 1998.

[3] P. Delacourt, and C. Wellekens, "DISTBIC: a speaker-based segmentation for audio data indexing," *Speech Communication*, Vol. 32, No. 1-2, pp. 111-126, 2000.

[4] L. Wilcox, F. Chen, D. Kimber, and V. Balasubramanian, "Segmentation of Speech Using Speaker Identification," *Proc. of ICASSP*, pp. I.161- I.164, 1994.

[5] S. Meignier, J.F. Bonastre, and S. Igounet, "E-HMM approach for learning and adapting sound models for speaker indexing," Proc. 2001: a Speaker Odyssey, pp. 175-180, 2001.

[6] R.B. Dunn, D. Reynolds, and T. Quatieri, "Approaches to Speaker Detection and Tracking in Conversational Speech," Digital Signal Processing, vol. 10, pp. 93-112, 2000.

[7] J. Pelecanos, S. Sridharan, "Feature warping for robust speaker verification," Proc. 2001: a Speaker Odyssey, pp. 213-218, 2001.

[1] This work was partially supported by the EU FP-6 IST Project DIVINES – Diagnostic and Intrinsic Variabilities in Natural Speech