

# A Hybrid Generative-Discriminative Approach to Speaker Diarization

Blind Submission

No Institute Given

**Abstract.** In this paper we present a sound probabilistic approach to speaker diarization. We use a hybrid framework where the number of speakers is estimated with a discriminative model. The output of this process is used as input in a generative model that can adapt to a novel test set and perform high accuracy speaker diarization. We manage to deal efficiently with the less common, and therefore harder, segments like silence and multiple speaker parts in a principled probabilistic manner.

## 1 Introduction

The objective of speaker diarization is to segment a digital recording in speaker-homogenous parts [11]. The output of speaker diarization can be used to improve sentence segmentation [1], and Automatic Speech Recognition (ASR) [7], thus consisting a very important step in analyzing multi-speaker digital recordings. The implementation of a robust system that performs automatic speaker diarization is a formidable task for three reasons. Firstly, in order to be applicable to novel digital recordings, we cannot assume the existence of any speaker-specific training data. Secondly, in order for our system to be robust in different scenarios we cannot assume knowledge of the microphone locations, or the existence of microphone arrays. Thirdly, if we want to use the output of the framework to improve ASR, we need to segment the digital recording with high precision - Cuendet et. al. in [12] showed that increasing the labeling precision by removing the low confidence parts from the training data will increase the ASR performance.

In recent research, there have been many different approaches to speaker diarization. In general, most approaches perform clustering in the audio descriptor space, and expect the clusters discovered to correspond to speaker-homogenous parts. The main problem then lies in dealing with silent parts and multiple-speaker situations, since they both appear for a very small fraction of the digital recording. For example, Laskowski and Schultz, in [7], assign segments of the audio stream to speakers through unsupervised clustering in the audio descriptors space. Using this initial assignment, they learn each person's voice model, and use feature space rotation or sample-level overlap synthesis to deal with multi-speaker parts. Angueral et.al. in [1], use a preprocessing step to exclude non-speech data from the stream, and use agglomerative clustering in the remaining data to assign each recording segment to the corresponding speaker. In this case, there is an assumption that all speech segments were generated by a single speaker.

In speech recognition or speaker identification, a person's voice is modeled in a generative way [4, 5, 9]. Furthermore, in [1] and [7], the speaker diarization is evaluated based on the improvement of the ASR, which is performed with a generative

model (Hidden Markov Model (HMM)/Gaussian Mixture Model (GMM)). Applying a generative model directly to speaker diarization is much harder.

In general we shall denote the number of participants with  $P$ , but for the moment, consider an example recording of two participants. Let's assume that the voice models are given and they are parameterized by  $\theta_1$  and  $\theta_2$ . An audio descriptor at time  $t$ ,  $(A_t)$ , can be generated from four different system states, corresponding to one, none, or both of the persons speaking. In general the system state space is of size  $2^P$ . We denote the system state on time  $t$  with  $x_t$ , which is a binary vector of length  $P$ . It is trivial to decide which person most probably generated  $A_t$ , by comparing  $p(A_t|\theta_p)$  for  $p = 1 : P$ . However, evaluating the system states is not that straightforward. In order to evaluate  $p(A_t|x_t)$  directly, we would need a state-specific distribution. However, some system states appear very rarely in our stream. Thus, we have too little data to learn the parameters of their distribution reliably.

A typical solution in this case is to assume that the state of each participant is independent of the state of the others. Although in a linguistic perspective this is not the case, from a signal processing side of view, it is a very realistic assumption. In our example, consider the state where the second person is speaking, this would be:

$$p(A_t|x_t) = \frac{p(x_t|A_t)p(A_t)}{p(x_t)} = \prod_{p:1..P} \frac{p(x_t(p)|A_t)p(A_t)}{p(x_t(p))} = \prod_{p:1..P} p(A_t|x_t(p)) \quad (1)$$

Even then though, we are left with the formidable task of estimating  $p(A_t|x_t(p))$  for  $x_t(p) = 0$ , which is the probability that a person generated this audio descriptor when they are **not** speaking. This is a quantity which we can not define directly from our data. In section 2 we describe a way to overcome this problem and perform generative speaker diarization, using a simple preprocessing step in which we estimate a distribution over the number of simultaneous speakers on each instant. In section 3 we test two different models for labeling time sequences in the task of determining the number of speakers on each time slice, a HMM and a Conditional Random Field (CRF). In section 4 we show the improvement on speaker diarization on audio streams coming from smart meeting rooms. In section 5 we conclude this paper with a short discussion of the proposed model and the results achieved.

## 2 Generative Modeling of Multiple Speakers

As described earlier, in a generative approach we need to estimate the probability that the audio descriptor  $A_t$  at time  $t$  was generated by each system state  $x_t$ . Namely we need to compute  $p(A_t|x_t)$  which is proportional to  $p(x_t|A_t)$ . In order to do this, we define a distribution over the number of speakers at time  $t$ ,  $p(n|A_t)$ . In this case, we

can rewrite  $p(x_t|A_t)$  as:

$$\begin{aligned}
p(x_t|A_t) &= \sum_n p(x_t, n|A_t) \\
&= \sum_n p(x_t|A_t, n) p(n|A_t) \\
&= \sum_n \frac{p(A_t, n|x_t)p(x_t)}{\sum_x p(A_t, n|x_t)p(x_t)} p(n|A_t) \\
&= \sum_n \frac{p(A_t|x_t)p(n|x_t)p(x_t)}{\sum_x p(A_t|x_t)p(n|x_t)p(x_t)} p(n|A_t)
\end{aligned} \tag{2}$$

for clarity of presentation we define as  $n_s(x_t)$  a function which returns the number of speakers implied by  $x_t$ . At this point, we notice that  $p(n|x_t)$  is 1 for  $n$  equal to  $n_s(x_t)$  and 0 otherwise. Thus we simplify equation 2 in:

$$p(x_t|A_t) = \frac{p(A_t|x_t)p(x_t)}{\sum_{x:n_s(x)=n} p(A_t|x_t)p(x_t)} p(n_s(x_t)|A_t) \tag{3}$$

Now, from equation 1 we can express  $p(A_t|x_t)$  as:

$$\begin{aligned}
p(A_t|x_t) &= \prod_p p(A_t|\theta_p)^{x_t(p)} Q^{1-x_t(p)} \\
&= [N - n_s(x_t)] Q \prod_{p:x_t(p)=1} p(A_t|\theta_p)^{x_t(p)}
\end{aligned} \tag{4}$$

where  $Q = p(A_t(i)|x_t(j))$  for  $x_t(j) = 0$ . Assigning the same  $Q$  for all speakers  $j$ , implies that all the persons affect the audio stream the same way when they do not speak, and it is a reasonable assumption. If we use uniform  $p(x_t)$  for all possible states we get:

$$p(x_t|A_t) = p(n_s(x_t)|A_t) \frac{\prod_{p:x_t(p)=1} p(A_t|x_t)}{\sum_{x:n_s(x)=n} \prod_{p:x_t(p)=1} p(A_t|x_t)} \tag{5}$$

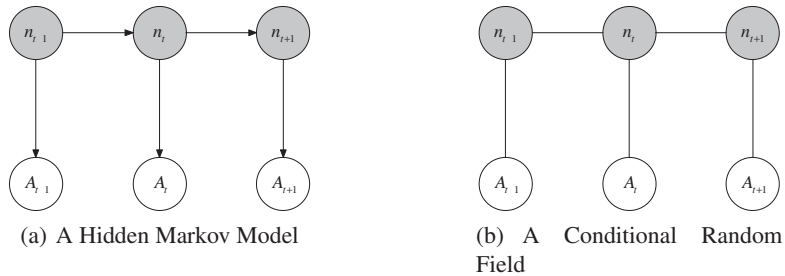
Intuitively, we have one unit of probability mass to distribute among the different possible system states. The states are grouped based on the number of speakers that each state implies, and the term  $p(n_s(x_t)|A_t)$  determines what part of the probability mass is allocated to each group. This part is then distributed among the states of the group through the second term of the product.

### 3 Modeling the Number of Speakers

The objective of our data preprocessing step is to estimate a distribution over the different numbers of speakers for each segment of the recording. Since the temporal patterns of the data are very important for such a task, we compared two probabilistic models that take into consideration the temporal relationships in our data. A generative one, in the form of a HMM [10], and a discriminative one, in the form of a linear CRF [6].

A HMM is a generative probabilistic model, defined fully by the prior probability of the system to be at each state on the first time slice of the model  $\pi$ ; the transition distribution  $p(n_t|n_{t-1})$  representing the probability of going from state  $n_{t-1}$  at time  $t-1$  to state  $n_t$ , and the observation model that defines the probability that an observation

IV



**Fig. 1.** The two models compared for estimating the number of speakers on each time slice. The grey nodes denote hidden variables, while the white nodes depict observable ones.

$A_t$  was generated by state  $n_t$ , namely  $p(A_t|n_t)$ . A graphical representation of a HMM can be seen in figure 1(a).

A linear chain CRF, parameterizes is a discriminative probabilistic model that resembles closely the HMM. As we can see in figure 1(b), the model still consists of a hidden variable and an observable variable at each time step. However, the arrowheads of the edges between the various nodes have disappeared, making this an undirected graphical model. This means that two connected nodes no longer represent a conditional distribution (e.g.  $p(A_t|n_t)$ ), but instead we speak of the potential between two connected nodes. This potential also represents the chance of observing a specific configuration of its variables, but unlike a probability is not restricted to be a value between 0 and 1.

The potential functions that specify the linear-chain CRF are  $\psi(n_t, n_{t-1})$  and  $\psi(n_t, A_t)$ . For clarity of representation these potential functions are written down using a more uniform notation, which allows different forms of CRFs to be expressed using a common formula. In this work, we adopt the notation of [13] and therefore define:  $\psi(n_t = i, n_{t-1} = j) = \lambda_{ijk} f_{ijk}(n_t, n_{t-1}, A_t)$  in which the  $\lambda_{ijk}$  is the parameter value (the actual potential) and  $f_{ijk}(n_t, n_{t-1}, A_t)$  is a feature function which in our case can be a binary indicator of whether  $n_t = i$  and  $n_{t-1} = j$  or it returns the value of the specific feature respectively. The index  $ijk$  is typically replaced by a one-dimensional index, so we can easily represent the summation over all the different potential functions.

The essential difference between HMMs and CRFs lies in the way we learn the model parameters. In the case of HMMs the parameters are learned by maximizing the *joint* probability distribution  $p(n_{1:T}, A_{1:T})$ . The parameters of a CRF are learned by maximizing the *conditional* probability distribution  $p(n_{1:T} | A_{1:T})$ . One of the main consequences of this choice, is that while learning the parameters of a CRF, we avoid modeling the distribution of the observations,  $p(A)$ . The conditional probability is modeled as:

$$p(n_{1:T}|A_{1:T}) = \frac{1}{Z(A)} \exp \left\{ \sum_{k=1}^K \lambda_k f_k(n_t, n_{t-1}, A_t) \right\} \quad (6)$$

HMM	0	1	2	3	4	CRF	0	1	2	3	4
0	<b>88.38</b>	6.79	0.543	2.71	1.56	0	<b>82.74</b>	10.05	0.95	3.73	2.51
1	23.98	<b>49.04</b>	9.17	13.17	4.61	1	18.07	<b>60.59</b>	14.6	5.91	0.81
2	10.32	14.94	<b>33.08</b>	17.32	24.32	2	2.64	25.40	<b>41.50</b>	21.8	8.55
3	15.89	23.43	16.03	<b>21.12</b>	23.50	3	2.71	8.49	24.86	<b>38.79</b>	25.13
4	4.07	9.17	13.17	16.6	<b>56.92</b>	4	3.66	3.05	4.75	20.58	<b>67.93</b>

**Table 1.** Cross validation results (%) acquired on determining the number of speakers

where  $Z(A)$  is the normalization function

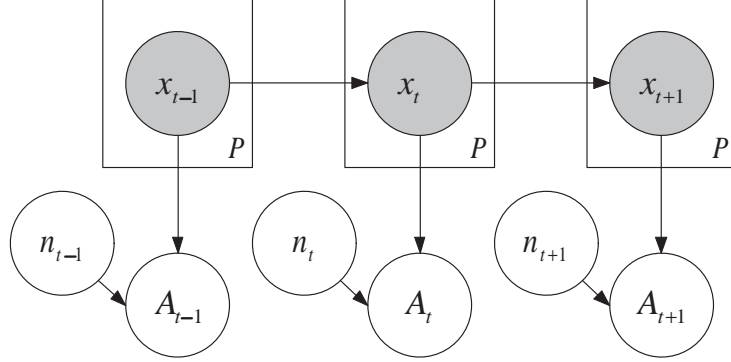
$$Z(A) = \sum_{n_t} \exp \left\{ \sum_{k=1}^K \lambda_k f_k(n_t, n_{t-1}, A_t) \right\} \quad (7)$$

Note that the quantity  $p(n_{1:T}, A_{1:T})$  is convex in the  $\lambda$ -space, and we can use any optimization algorithm to obtain the optimal  $\lambda$  values. In our implementation we used the BFGS algorithm [2] which was shown to be the most efficient approach [13]

We trained and tested both our models on 2 hours of data, coming from multiple speakers. In this corpora, coming mainly from interviews and video-conferences, the number of speakers on each time window was set manually. We extracted the 13 first Mel frequency cepstra coefficients, and concatenated their first and second order differences. Thus, a 39 dimensional vector was created for each audio window, while the stream was segmented in 25 ms audio windows with 10 ms overlap. The HMM modeled the generative distribution for different numbers of speakers as a Gaussian mixture model of 15 components with diagonal covariances. The CRF used one feature function per dimension which returned the observation value itself. We selected the pre-processing model for our final framework using 10-fold cross validation on this data. The results achieved under both models can be seen in table 1.

The CRF performed much better than the HMM, and therefore it was preferred for our final implementation. The main diagonal of the confusion matrixes contains the accuracy per class, and CRF was able to distinguish multiple speaker states much better. It is interesting to notice that the two diagonals near the central one, highlighted with a gray background in table 1, contain the main mass of classifications. This is reasonable, since when one person is speaking there are long pauses, and while two or more people are speaking it is often the case that only 2 of them are active over such a small time window of 25ms.

Finally, it is important to notice that the number of speakers on a time slice can be estimated using parameters trained with speaker-independent data. Therefore, a discriminative approach, like the CRF, which requires labeled data, can also be used. In actual speaker diarization, described in section 4, we require speaker-specific voice models, without assuming the existence of any labeled speaker-specific data. Thus, we can only use a generative approach, and learn the parameters of each person’s model directly from our test data, using the Expectation Maximization algorithm [3].



**Fig. 2.** The model used for learning and inference during speaker diarization. Notice that node  $X(p)$  is repeated  $P$  times at each time slice. The state of different persons at a specific time slice  $t$  ( $X_t(p)$ ) are interdependent since they are parents of the same observable node  $A_t$ , but they are independent at the transition phase.

#### 4 Inference for Speaker Diarization

The distribution on the number of speakers of each time-slice, acquired from the pre-processing step, is used as observation in our speaker diarization model. The graphical representation of the proposed probabilistic model can be seen in figure 2 and comes in the form of a HMM. The generative choice here is necessary, since we want to learn the parameters of our model directly from our training data.

In the proposed approach, we assume a recording containing  $P$  persons. Therefore, the possible system states take values from a discrete space of  $2^P$  values. We represent these states as binary vectors, with the  $p^{th}$  element being 1 if the corresponding person is speaking and 0 otherwise. We assume a uniform prior probability for the system being in any state at the beginning of the stream. The transition matrix,  $A$ , is of size  $2^P \times 2^P$ , with the element  $A_{ij}$  corresponding to the probability  $p(x_t = j | x_{t-1} = i)$ . This creates a very number of parameters ( $2^P(2^P - 1)$ ), with some specific state transitions becoming very improbable, or even never appearing in a given recording. We reduce the number of necessary parameters to  $2P$  assuming that the transition of each person's state is independent of the others. In principle, humans perceive the changes in the state of their co-speakers and act based on this, but in practice this simplification works well. Thus we denote with  $A_p^1$  the probability that person  $p$  will remain speaking in a system transition ( $p(x_t(p) = 1 | x_{t-1}(p) = 1)$ ), and  $A_p^0$  that person  $p$  will remain silent, and we get:

$$A_{ij} = p(x_t = j | x_{t-1} = i) = \prod_p p(x_t(p) | x_{t-1}(p)) \quad (8)$$

where we need to learn  $2P$   $A_p$  parameters. Finally, the observation model defines the probability that a specific observation was generated from a given system state,  $p(A_t | x_t)$ . We acquire this using Bayes Rule:

$$p(A_t | x_t) = \frac{p(x_t | A_t) p(A_t)}{p(x_t)} \quad (9)$$

and assuming uniform priors for all different system states.

## 5 Learning with the E.M.

The framework presented so far is generic, in the sense that any voice model can be incorporated to model  $p(A_t|x_t(p) = 1)$ . In our implementation we modeled each participants voice with a 15 component Gaussian mixture model in the feature space. In order to perform speaker diarization as described in section 4, we need to acquire the voice model and the transition probabilities for each person. We acquire these parameters using the E.M. algorithm [3] directly on the test data.

In the E-step, we estimate the expectation of the system to be at a specific state on each time slice. We perform the forward procedure, which estimates  $\alpha_i(t) = p(A_{1..t}, x_t = i)$  and the backward procedure that estimates  $\beta_i(t) = p(A_{t+1..T}|x_t = i)$  for all time slices. These quantities can be computed efficiently using a recursive formula, more details of which can be found in [3]. We can now estimate the probability of the system to be in state  $i$  at  $t$ ,  $\gamma_i(t) = p(x_t = i|A_{1..T})$  as:

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{2^P} \alpha_j(t)\beta_j(t)} \quad (10)$$

as well as the probability of the system having a transition from state  $i$  to state  $j$  at time  $t$ ,  $\xi_{ij}(t) = p(x_t = i, x_{t+1} = j|A_{1..T})$  as:

$$\xi_{ij}(t) = \frac{\gamma_i(t)A_{ij}p(A_{t+1}|x_t = j)\beta_j(t+1)}{\beta_i(t)} \quad (11)$$

In our case, the state of each person is independent of the others. Thus, the probability of a person to be speaking on a specific time slice is  $\gamma^p(t) = p(x_t(p) = 1|A_{1..T})$ :

$$\gamma^p(t) = \sum_{i:x_t(p)=1} \gamma_i(t) \quad (12)$$

and the probability of a person transition from state  $k$  to  $l$ , denoted as  $\xi_{kl}^p(t)$

$$\xi_{kl}^p(t) = \sum_{i:x_t(p)=k, j:x_{t+1}(p)=l} \xi_{ij}(t) \quad (13)$$

In the m-step we are going to use these expectations to set the model parameters to the values that maximize the complete-data likelihood. For each person:

$$A_p^0 = \frac{\sum_T \xi_{00}^p}{\sum_T 1 - \gamma^p(t)} \quad A_p^1 = \frac{\sum_T \xi_{11}^p}{\sum_T \gamma^p(t)} \quad (14)$$

which correspond to the expectation of person  $p$  to remain silent or speaking.

The voice model of each person is modeled as a Gaussian mixture model of 15 components. For each component  $c$  we need to estimate the mean  $\mu_c^p$ , covariance  $\Sigma_c^p$  and mixture proportion  $\pi_c^p$ . If we denote with  $p_c^p(t)$  the probability that observation  $A_t$  was generated from the  $c^{th}$  component of the  $p^{th}$  person, then

$$p_c^p(t) = \gamma^p(t) \frac{\mathcal{N}(A_t; \mu_c^p \Sigma_c^p)}{\sum_c \mathcal{N}(A_t; \mu_c^p \Sigma_c^p)} \quad (15)$$

where  $\mathcal{N}$  denotes the Gaussian kernel. The M-step equations become:

$$\begin{aligned}\mu_c^p &= \frac{1}{N_c} \sum_t p_c^p(t) A_t \\ \Sigma_c^p &= \frac{1}{N_c} \sum_t p_c^p(t) (\mu_c - A_t(i))^2 \\ \pi_c^p &= \frac{N_c}{\sum_c N_c}\end{aligned}\tag{16}$$

where  $N_c = \sum_t p_c^p(t)$ . Note here that  $(\mu_c - A_t(i))^2$  represents raising element-wise the result in the power of 2, leading to spherical covariance matrices.

## 6 Results

We applied the proposed model to perform speaker diarization in a 30 minutes audio recording, coming from a smart meeting room. The recording was part of the CHIL dataset, and it was used in the CLEAR evaluation contest (<http://www.clear-evaluation.org/>). The specific audio recording comes from the IDIAP smart meeting room [8]. The objective of the experiment was threefold. First, we are interested in the accuracy of the CRF in detecting the correct number of speakers for each time slice. Second, we wanted to see how well the generative approach described in section 2 can handle the parts that multiple speakers vocalize. Thirdly, we want to see how much this improves the speaker diarization results in comparison with approaches that assume a single-speaker per time slice, or a single-speaker after removing the silence parts.

Initial, in section 6.1, we present the results on detection of the number of speakers. In section 6.2, we present the results of three speaker diarization experiments on our data. In the first one, our proposed model is applied and results acquired through this model are labeled as *full*. We then lowered the dimensionality of our hidden states space, to that including only states implying a single speaker or silence. We label this experiment as *low*. Finally, we excluded the audio windows labeled from our CRF as silence, and used a model containing only the single speaker states. This experiments are denoted as *pre*. In the *low* experiments, we did not use the distribution over the number of speakers but instead modeled silence as a fifth speaker. In the *pre* experiments, the distribution over the number of speakers would not make any difference since all states correspond to a single speakers. In section 6.3 the results on classification of multiple speaker parts are detected.

### 6.1 Accuracy in detection of number of speakers

The results of speaker diarization using the speaker independent data can be seen in table 6.1. As we can see, the CRF exhibits similar behavior with that in table 1, where it manages to distinguish the different number of speakers reliably. Furthermore, it distinguishes very well between silence and single speaker windows, which is very important since these are the dominant classes of a meeting recording. The CRF is not a Bayesian model. Therefore, the conditional probability  $p(n_{1:T}|A_{1:T})$  that we maximize, is not a posterior but a conditional likelihood function. In order to focus the classification accuracy on silence and single speaker data, we had to train our model with a dataset containing more data from the specific classes.

<i>Number of Speakers</i>					
<i>Accuracy</i>	0	1	2	3	4
0	<b>0.52</b>	0.43	0	0.02	0.01
1	0.018	<b>0.96</b>	0.0	0.01	0.01
2	0.01	0.75	<b>0.19</b>	0.03	0.01
3	0	0.74	0.11	<b>0.14</b>	0
4	0	0.17	0	0	<b>0.82</b>

**Table 2.** Number of speakers detection accuracy on the audio recording coming from the IDIAP smart meeting room data.

## 6.2 Accuracy in speaker diarization

In table 6.2 we can see the results achieved under the three different assumptions. In *pre*, we assume that each window was created by a single speaker. As a consequence, the silence parts are also assigned to a speaker. The segments belonging to a single speaker have high classification accuracy, since there are no multi-speaker or silent system state labels available to the model. The random classification accuracy here would be 25%. In *low*, only silence and single speaker states compete. We achieve here high accuracy results in silence detection, but lower on the speaker diarization. When a single person is speaking there are silent parts, and therefore segments belonging to a single speaker are classified as silence. In *full*, we can see the results of our proposed framework. Silence detection has slightly lower precision, since silence is not modeled as an independent speaker, but rather detected through the preprocessing step. On the other hand, the speaker diarization has much higher accuracy, and this difference can prove essential in ASR or automatic transcription tasks.

## 6.3 Multiple Speaker parts

Finally, we would like to investigate is how well our model classifies the multi-speaker parts. The results are visible in table 6.3 and follow the pre-processing results presented in table 6.1. Thus, the parts with two or three speakers are detected with low accuracy, while the four person parts are detected very accurately. The reason is that most of the

<i>Pre</i>	0	1	2	3	4	<i>Low</i>	0	1	2	3	4	<i>Full</i>	0	1	2	3	4
0	0.00	0.67	0.13	0.12	0.07	0	<b>0.79</b>	0.07	0.05	0.05	0.03	0	<b>0.72</b>	0.14	0.02	0.08	0.03
1	0.00	<b>0.40</b>	0.21	0.18	0.20	1	0.11	<b>0.35</b>	0.18	0.15	0.18	1	0.10	<b>0.41</b>	0.15	0.13	0.18
2	0.00	0.23	<b>0.46</b>	0.14	0.14	2	0.14	0.17	<b>0.44</b>	0.12	0.13	2	0.17	0.13	<b>0.48</b>	0.12	0.08
3	0.00	0.19	0.13	<b>0.48</b>	0.18	3	0.14	0.11	0.15	<b>0.44</b>	0.16	3	0.12	0.09	0.10	<b>0.54</b>	0.14
4	0.00	0.11	0.08	0.13	<b>0.66</b>	4	0.09	0.07	0.06	0.11	<b>0.64</b>	4	0.09	0.07	0.06	0.11	<b>0.63</b>

**Table 3.** Speaker diarization results

two person speaking results correspond to audio feedback given from one person when another person speaks. Thus, they are short in duration and harder to detect.

<i>Number of speakers in a window</i>	Total windows in data	Windows detected	Speakers detected
2	5300	0.28	0.54
3	1472	0.22	0.92
4	2693	0.65	1.00

**Table 4.** Accuracy in multiple speaker parts.

In the third column of table 6.3, *speakers detected*, we see how many of the correctly detected multi-speaker windows were assigned to the correct persons. In the case of two speakers there are 6 difference system states combinations to choose from, in the case of three speakers there are 4 states, while in the case that four speakers are detected there is only one corresponding system state. This is an indication of the difficulty of selecting the correct speakers and it is depicted in the *speakers detected* results, where the accuracy increases as the number of states to choose from decreases.

## 7 Discussion & Conclusions

The results in section 6 present the potential of a hybrid approach to speaker diarization. We do not claim that our model and parameter choices for each sub-task are optimal, but it is our strong belief that the proposed method consists a sound probabilistic approach to the task of speaker diarization. The experimental results of section 6.1 show that a discriminative approach can detect the number of active speakers reliably. Multi-speaker parts, although rear in the stream, can be detected without jeopardizing high accuracy in single-speaker and silence detections. The CRF framework allows much space for tuning, and the use of more training data, coming straight from meeting audio in combination with the use of problem-specific features can further improve the results.

The results in section 6.2 show that the proposed hybrid model can rest the assumptions about a single speaker (and silence) at each time of the stream. These two classes are detected with extremely high accuracy, while the multi-speaker parts are treated in a uniform manner. Once more, more elaborate voice-models or voice-synthesis methods can improve the results further. Finally in section 6.3, we present the results on the detected multi-speaker parts. We see that the audio modality is partially able to distinguish the active speakers. In order to improve results here, either the video modality or a different voice-synthesis model should be used.

## References

1. Xavier Anguera, Chuck Wooters, and Javier Hernando. Automatic cluster complexity and quantity selection: Towards robust speaker diarization. In Steve Renals, Samy Bengio, and Jonathan G. Fiscus, editors, *MLMI*, volume 4299 of *Lecture Notes in Computer Science*, pages 248–256. Springer, 2006.
2. D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.
3. J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, 1997.
4. John Dines and Mathew Magimai Doss. A study of phoneme and grapheme based context-dependent asr systems. IDIAP-RR 12, IDIAP, 2007.
5. Martin Karafiát, František Grezl, Petr Schwarz, Lukáš Burget, and Jan Černocký. Robust heteroscedastic linear discriminant analysis and lrcr posterior features in meeting data recognition. *Lecture Notes in Computer Science*, 2006(4299):275–284, 2006.
6. John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
7. Kornel Laskowski and Tanja Schultz. Modeling vocal interaction for segmentation in meeting recognition. *Lecture Notes in Computer Science*, 4892:259–270, February 2008.
8. D. Moore. The IDIAP Smart Meeting Room, 2002.
9. Darren Moore, John Dines, Mathew Magimai Doss, Jithendra Vepa, Octavian Cheng, and Thomas Hain. Juicer: A weighted finite-state transducer speech decoder. IDIAP-RR 21, IDIAP, 2006. To appear in *MLMI'06*, Washington DC.
10. Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296, 1990.
11. D. Reynolds and P. Torres-Carrasquillo. Approaches and applications of audio diarization. In *IEEE ICASSP*, page 953956, 2005.
12. Dilek Hakkani-Tur Sebastien Cuendet and Elizabeth Shriberg. Automatic labeling inconsistencies detection and correction for sentence unit segmentation in conversational speech. *Lecture Notes in Computer Science*, 4892:144–155, February 2008.
13. Charles Sutton and Andrew McCallum. *Introduction to Statistical Relational Learning*, chapter 1: An introduction to Conditional Random Fields for Relational Learning, page (Available online). MIT Press, 2006.