IDIAP RESEARCH REPORT

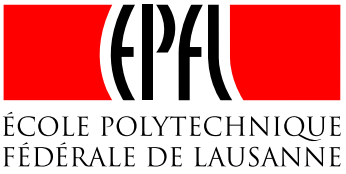
NOVEL INITIALIZATION METHODS FOR SPEAKER DIARIZATION

David Imseng

Idiap-RR-07-2009

MAY 2009

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch



Novel initialization methods for Speaker Diarization

David Imseng
david.imseng@gmail.com

Supervisors:
Prof. Hervé Bourlard and Dr. Gerald Friedland

March 11, 2009
Master thesis in Communication Systems
Swiss Federal Institute of Technology in Lausanne

Abstract

Speaker Diarization is the process of partitioning an audio input into homogeneous segments according to speaker identity where the number of speakers in a given audio input is not known a priori. This master thesis presents a novel initialization method for Speaker Diarization that requires less manual parameter tuning than most current GMM/HMM based agglomerative clustering techniques and is more accurate at the same time.
The thesis reports on empirical research to estimate the importance of each of the parameters of an agglomerative-hierarchical-clustering-based Speaker Diarization system and evaluates methods to estimate these parameters completely unsupervised. The parameter estimation combined with a novel non-uniform initialization method result in a system that performs better than the current ICSI baseline engine on datasets of the National Institute of Standards and Technology (NIST) Rich Transcription evaluations of the years 2006 and 2007 (17% overall relative improvement).

Acknowledgments

This Master thesis was done in the context of an internship at the International Computer Science Institute (ICSI), Berkeley, through an exchange program jointly sponsored by Swiss NSF through the National Center of Competence in Research (NCCR) on "Interactive Multimodal Information Management" (IM2, www.im2.ch) and the European Integrated Project on "Augmented Multiparty Interaction with Distance Access" (AMIDA, www.amidaproject.org).
Today, I can look back at six great months in Berkeley and therefore I want to thank some people. First, I would like to thank Dr. Gerald Friedland, for supervising me at the International Computer Science Institute and always having time to discuss my questions. He also helped me quite a bit with the writing and polished my English. I would also like to thank Prof. Hervé Bourlard who was my supervising professor at EPFL.
The working environment at ICSI is one of the most comfortable and inspiring ones I have seen so far. I appreciated working at that place and would like to thank everybody from the Speech group for the fruitful conversations during lunch and tea and for the relaxing foosball games.
Furthermore, I want to thank Erika who has drawn the comic for the introduction.

Contents

Chapter 1

Introduction

> Machines will be capable, within twenty years, of doing any work that a man can do.
>
> Herbert Simon, 1965[1]

A fundamental goal of Artificial Intelligence is to design machines that are able to imitate human skills. The telephone switchboards for example, used by the so-called operators to manually connect a group of telephones, disappeared during the last half of the 20th century and robots in serial production facilities replace human workers and accomplish the task faster and cheaper these days. Interestingly, in more intellectual tasks, like playing chess, computers are able to win against human players, but it is the much less sophisticated tasks such as walking or having a conversation that are the most challenging for computers. One example of such a computer-challenging task is Speaker Diarization which aims at detecting *who spoke when*[2]. A task that is almost trivially solved by human beings in every conversation. Identifying the number of speakers as well as the speaker turns in a meeting recording turned out to be a very demanding task for computer systems.
Knowing when each speaker is speaking in a meeting recording is useful as a pre-processing step in automatic speech recognition systems for instance, to improve the quality of the output. Such pre-processing may include vocal tract length normalization and/or speaker adaptation [Ajmera, 2003]. Automatic speaker segmentation may also be useful in information retrieval and as part of the indexing information of audio archives.
A typical meeting situation is shown in Figure 1.1. People are discussing a certain topic. There are several sources of background noise like the clock, somebody knocking at the door, cellphones ringing and so on. The task of Speaker Diarization may be divided into two subtasks (see Figure 1.2). A speech/non-speech detector needs

[1] Quoted by Crevier 1993 in *The Tumultuous Search for Artificial Intelligence*, p. 109

[2] The the term diarization is not used very often outside the speech community. The word is related to diary, indicating an annotation of events with timemarks [Leeuwen and Konečný, 2008].

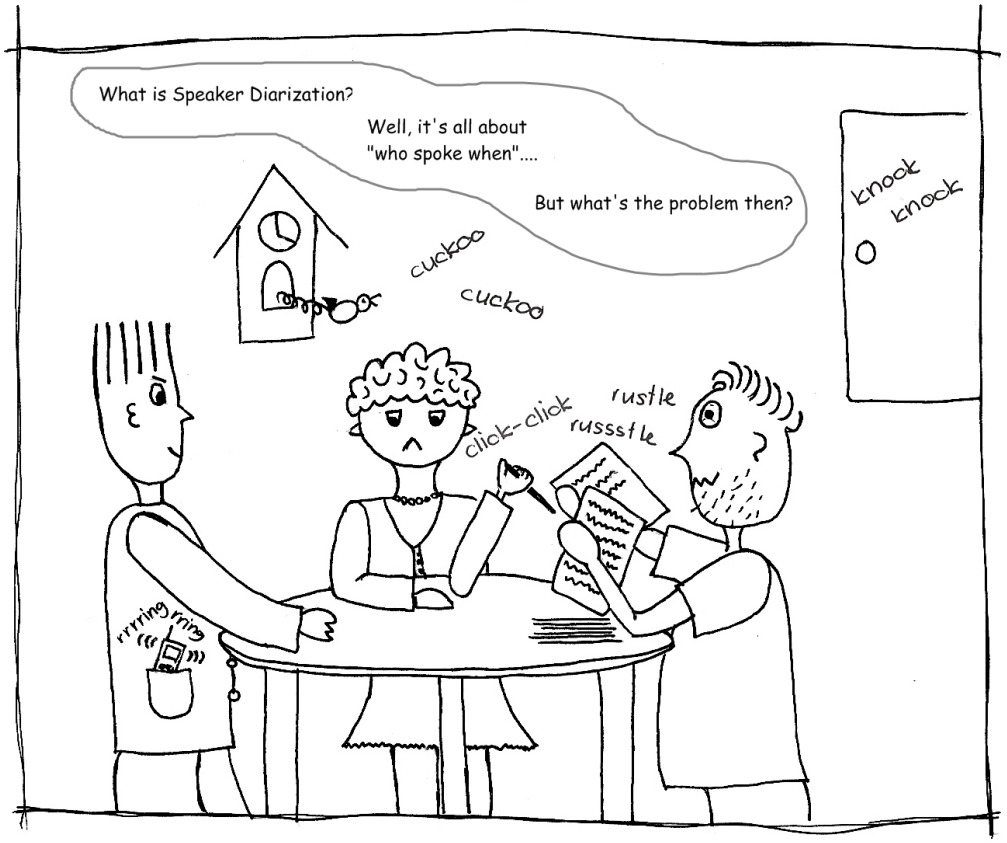


Figure 1.1: A typical Meeting Situation. People are discussing about a certain topic and there are several sources of background noise.

to filter out background noise and non-speech to extract the speech parts from the recordings. Having identified the speech regions (the yellow segments in Figure 1.2), a clustering algorithm merges all the segments from the same speaker together and finally answers the question *who spoke when* (segmentation of the speech regions can be seen at the bottom of Figure 1.2). In the example meeting, presented in Figure 1.1, the clustering algorithm needs to recognize that there are three speakers and assign each speech part to the corresponding speaker.
This master thesis is structured as follows: in Chapter 2, some related work is briefly summarized followed by a description of previous work that was done at the International Computer Science Institute (ICSI) in Chapter 3. In Chapter 4, the problem is stated and the goals of this work are argued. After that, Chapter 5 and 6 present methods to solve the explained problem and also include results of the experiments to test the proposed methods on different data-sets. Some limits are identified in Chapter 7, then a conclusion is drawn and future work is proposed.

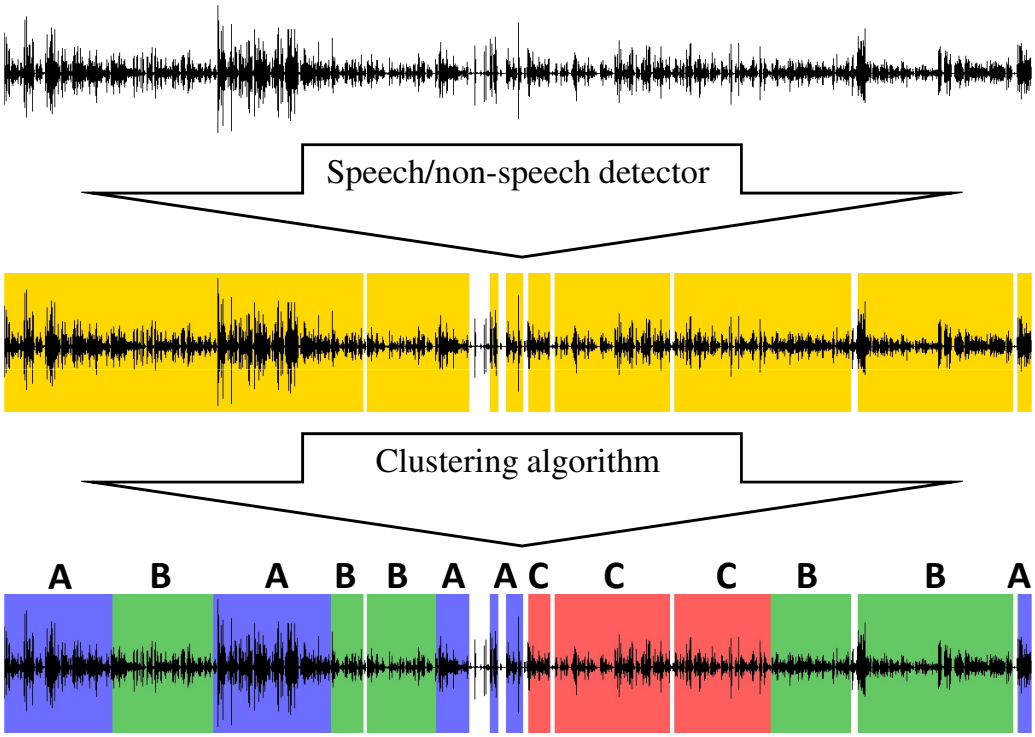


Figure 1.2: A schematic overview over the Speaker Diarization process that can be divided into two subtasks: Speech/non-speech detection and a clustering algorithm. At the top, an audio signal is displayed, in the middle, the speech regions are marked in yellow and at the bottom, different speaker regions are marked with different colors.

Chapter 2

Related work

A large amount of work has been done in the field of Speaker Diarization. This chapter will briefly present a selection of approaches. Most techniques participated in the National Institute of Standards and Technology (NIST[1]) Rich Transcription (RT) evaluation, hold last time in spring of the year 2007. NIST distinguishes between recordings with multiple distant microphones (MDM) and recordings with one single distant microphone (SDM). In the case of MDM, typically beamforming is performed to produce one single channel out of all the available ones. After that, most approaches apply a Speech Activity Detection (SAD), followed by some clustering algorithm that produces the final output of the Speaker Diarization engine.

2.1 Agglomerative Clustering

The speaker diarization engine of the International Computer Science Institute (ICSI) is based on agglomerative clustering. That system is described in more detail in Chapter 3. The speaker diarization engine of the University of Catalonia (UPC) is also based on agglomerative clustering [Luque et al., 2008] and is similar to the one from ICSI but performed much worse at the NIST RT-07 evaluation, therefore the system is not presented here in more detail.
The LIMSI RT-07 speaker diarization system uses agglomerative clustering in combination with speaker identification techniques [Zhu et al., 2008]. After a Log-Likelihood Ratio (LLR) based speech activity detection (acoustic models for speech and silence both consist of 256 Gaussians and were trained on about 2 hours of lecture data), the Bayesian Information Criterion (BIC) based clustering, is combined with Speaker Identification (SID) techniques. BIC is used for the inter-cluster distance measure and the stop criterion. The penalty term (see glossary), which is weighted by a manually tuned parmater, is locally computed which was shown to outperform the globally calculated one [Barras et al., 2004]. After the agglomerative clustering, state-of-the-art speaker identification techniques are used to improve the

[1]Many acronyms are explained in more detail in the glossary at the end of this thesis

quality of the speaker clustering. A Universal Background Model (UBM) composed of 128 Gaussians (trained on a few hours of broadcast data) is used for that purpose. This diarization engine had the lowest Speech Activity Detection Error rate for conference data among all RT-07 participants but the highest Speaker Error rate.
Another approach using agglomerative clustering is the AMIDA submission to the RT-07 evaluation. One of the goals of this approach is to have as little tunable parameters as possible [Leeuwen and Konečný, 2008]. Basically the system uses PLP coefficients and log energy. The SAD is done with a two-state HMM, both states (speech and silence) are modeled with diagonal covariance GMMs made up of 16 Gaussians. An initial segmentation is done with the BIC system described in [Leeuwen, 2005]. The system was manually tuned to tend to over-segment and under-cluster and ends up with typically 40 clusters. The approach uses two sets of GMMs, one with a flexible number of Gaussians per cluster (to have about 4.8 seconds of data for each Gaussian) for the Viterbi segmentation and one with 64 Gaussians per cluster for determining the clusters to merge and the stop criterion (the same number of Gaussians is used, to get rid of the penalty term in the BIC comparison). Even so this approach is computationally very expensive, the performace is relatively poor and in spite of declaring parameter reduction as an explicit goal, the system to do the initial segmentation was manually tuned, the newly created parameter *number of Gaussians per cluster* to determine the clusters to merge and the stop criterion was set to 64 and the 4.8 seconds of data that is needed for each Gaussian is an empirically determined value.

2.2 Direction of Arrival Estimate and Acoustic Feature Information

In [Koh et al., 2007] the contribution of I^2R-NTU to the NIST RT-07 evaluation is presented. This approach only works if multiple audio channels are available (MDM condition). At first, the direction of arrival (DOA) is estimated, which is used to do a bootstrap clustering afterwards. The clustering is then followed by a cluster purification that makes use of MFCC acoustic features which are extracted from the (with the use of beamforming) enhanced recording. After having applied a voice activity detector to retain only the high energy frames, non-speech and silence is removed. In contrast to other approaches, this one is applying the SAD after the clustering.
The bootstrap clustering only works well if the speakers do not move during the meeting, but the numbers of clusters that were found for the NIST RT-07 tasks was quite accurate. Unfortunately, this approach contains parameters that have to be

carefully tuned respectively models that have to be trained on a training set that is supposed to be similar to the test set. The DOA estimation uses a parameter that was adjusted, the model-based non-speech removal uses a threshold that was tuned on a training set and the silence removal also depends on an optimized threshold.

2.3 Evolutive Hidden Markov Model (E-HMM)

The LIA submission for the RT-07 evaluation is based on evolutive Hidden Markov Models [Fredouille and Evans, 2008]. This approach uses Linear Frequency Cepstrum Coefficients (LPCCs). After the SAD, during which the audio is modeled with a two states HMM (one state for speech and one for non-speech), the clustering process starts with an HMM containing one single state (world model) modeled with 128 Gaussians, trained on the whole audio chunk. A pre-segmentation determines an initial segmentation based on a classical Generalized Likelihood Ratio (GLR) criterion-based speaker turn detection that uses a manually tuned threshold. An at least three seconds long segment is picked from the initial segmentation according to a likelihood maximization criterion, a speaker model is trained on the picked segment and then the model is added to the HMM as new state. The audio is iteratively resegmented with an adaption/decoding loop before it is decided if the newly added speaker is relevant or not (according to some heuristics, that imply at least one parameter). The algorithm stops if there are no more at least three seconds long segments left in the initial segmentation.
This approach performed relatively well on the SDM task, but was not really able to benefit from the additional information available in the MDM task. In addition to the explicit parameters as the tuned threshold used during the speaker turn detection and the heuristic rules, the number of Gaussians (128) and the length of the segments (three seconds) may also be considered as manually tuned parameters.

2.4 Information Bottleneck (IB) principle

The Information Bottleneck approach is different from all the other approaches presented so far. It is a non-parametric framework that does not use an explicit speaker model for every cluster. In general, the GMM estimation for every cluster is computationally expensive, thus this approach is faster than the other approaches. Information Bottleneck clustering is based on preserving the relevant information specific to a given problem [Tishby et al., 1999]. IB tries to find the trade-off between the most compact representation and the most informative representation of the data what corresponds to the maximization of the following criterion [Vijayasenan

et al., 2008a]:

$$\Gamma = I(Y, C) - \frac{1}{\beta} I(C, X) \tag{2.1}$$

where X is a set of elements, C is a set of clusters, Y is set of variables of interest and β is the Lagrange multiplier representing the trade off between the amount of information preserved $I(Y, C)$ (mutual information between Y and C) and the compression of the initial representation $I(C, X)$ (mutual information between C and X).

In [Vijayasenan et al., 2008a], two different IB techniques are described and the IB principle is applied to Speaker Diarization. The first technique is the Agglomerative Information Bottleneck technique that basically starts with one cluster per data point and then merges clusters until the desired number of clusters is obtained. At every step, the merging decision is taken to minimize the decrease of the objective function, thus the agglomerative Information Bottleneck technique is a greedy approach and based on a local criterion. Therefore it produces only an approximation of the optimal solution. The other technique, Sequential Information Bottleneck tries to improve the objective function in a given partition of the set into K clusters, where K is a fixed number. Speaker Diarization is performed by combining agglomerative and sequential techniques and different model selection techniques that are needed because the number of speakers in the task is not known a priori. The parameter β was manually tuned and the better performing model selection criteria, Normalized Mutual Information (NMI), makes the use of a manually tuned threshold value.

This approach did not participate in the NIST RT-07 evaluation, but it is shown that the results are equivalent to the ones from current HMM/GMM-based baseline systems [Vijayasenan et al., 2008b].

Chapter 3

Previous work at ICSI

The International Computer Science Institute (ICSI) is very active in the domain of Speaker Diarization. The institute participated several times in the National Institute of Standards and Technology (NIST) Rich Transcription (RT) evaluations. The last evaluation was hold in 2007 where ICSI's engine outperformed all other participating teams and was the only participant being able to get a Speaker Diarization Error Rate (DER) below 10% [1]. This system represents the current baseline and it is briefly described in this chapter (for a more detailed description see [Wooters and Huijbregts, 2007]). A schematic overview of the system is shown in Figure 3.1.

3.1 Front-end Acoustic Processing

The acoustic processing of the baseline system consists of three steps. All three steps are listed in this section and very briefly described.

1. **Wiener Filtering** The implementation of the Wiener Filter is an adapted version of the noise reduction algorithm developed for the Aurora 2 front-end ([Wooters and Huijbregts, 2007] and [Adami et al., 2002]).

2. **Beamforming** After the Filtering, a single channel is created by running delay and sum beamforming on the separate channels. The BeamformIt 2.0 toolkit is used for that purpose [Miró, 2006]. (In case of Single Distant Microphone (SDM) recordings, the second step cannot be applied to the data.)

3. **Feature extraction** The baseline system uses two different types of features: Mel Frequency Cepstrum Coefficients (MFCC) and delay features. Obviously, the second feature stream is only used if multiple audio channels are available. The MFCCs are created using the HTK toolkit [HTK, 2007] with a 10 ms frame rate and a 30 ms analysis window. The delay features are calculated with a frame rate of 10 ms, and an analysis window of 500 ms. The

[1] http://www.nist.gov/speech/tests/rt/2007/workshop/RT07-SPKR-v7.pdf

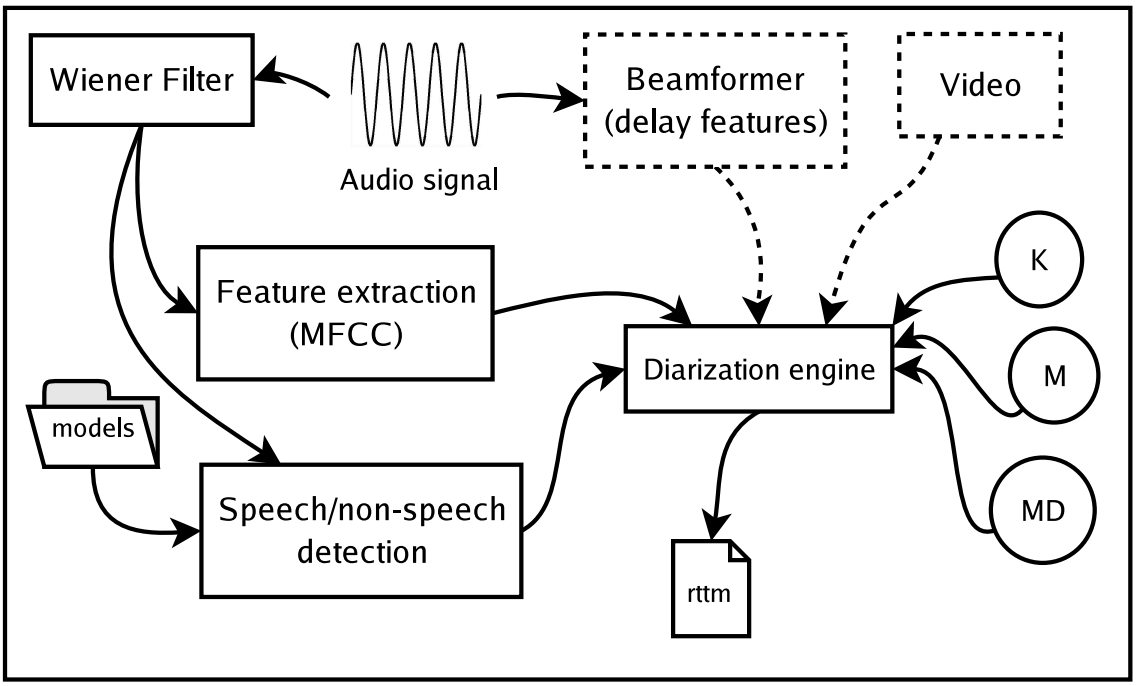


Figure 3.1: Schematic overview of the ICSI Speaker Diarization engine. Rectangular boxes are different components of the system. The delay features may only be used in the MDM case and video features when videos are available. Both feature types are not used during this work but successfully combined with acoustic features at ICSI. Therefore these boxes are dashed. The Speech/non-speech detector takes a pre-trained model as input (see Section 3.2) and the Diarization engine has the following parameters: the number of initial clusters K, the number of Gaussians per initial cluster M and the model constraint MD which is the minimal duration of a speech segment (see Section 3.3). The output of the Diarization engine is a Rich Transcription Time Marked file (rttm, [NIST, 2009]).

Probability Density Function (PDF) used in the diarization system is modeled with a Gaussian Mixture Model (GMM). If multiple feature streams are available, separate GMMs are used for different kind of features and the PDF is a weighted combination of the GMMs of the different feature types. The weighting is done automatically by the algorithm described in Chapter 5.3.2 *Automatic Features Weight Estimation* of the PhD thesis of Xavier Anguera Miró [Miró et al., 2006]. Recently, the conventional audio features were also successfully combined with video features [Friedland et al., 2009a]. However, for this work, the single stream engine, only using MFCC features, is used.

3.2 Speech/non-speech detection

The speech/non-speech detector *SHOUT* [Huijbregts, 2008] takes an audio file, some pre-trained models and a UEM file (un-partitioned evaluation map, [NIST, 2009]) as input. The latter causes that speech/non-speech detection is only done on the parts that are defined in the file. For the speech/non-speech detection, three different classes are used: speech, silence and audible non-speech. The detector consists of three steps: At first, the initial segmentation is created with a Hidden Markov Model (HMM) that contains two Gaussian Mixture Models (GMM), one for speech and one for silence. These models were trained on broadcast news data in advance. In the second step, the silence region is split into regions with low energy and regions with high energy. Then, a GMM containing 7 Gaussians is trained on the low energy regions, a GMM with 18 Gaussians on the high energy regions and a GMM with 24 Gaussians on the speech regions. The first and second step assumes that there is audible non-speech in every recording, what might be inaccurate. Therefore, at the third step this assumption is tested and if the models for speech and audible non-speech are too similar, they are merged. The Bayesian Information Criterion is used to check whether it is better to model speech and audible non-speech with one combined or two separate models. Finally, the output of the speech/non-speech detector is a file in RTTM (Rich Transcription Time Marked, [NIST, 2009]) format.

3.3 Agglomerative clustering

The clustering engine takes the output of the speech/non-speech detector and features as input. The acoustic data is modeled with an ergodic Hidden Markov Model (HMM) where every state represents a cluster and is divided into a certain number of sub-states, imposing a minimum duration (MD) constraint on the model. Initially the data is equally split into K clusters under the assumption that the number of speakers is not known a priori, but even so, K needs to be larger than the number

of speaker that are actually present in the data to process. Every cluster is modeled with a GMM consisting of M Gaussians. These three parameters (MD, K and M) are part of the input and need to be properly chosen.
Then the algorithm performs an agglomerative clustering which can be seen as an iterative training-segmenting-merging procedure:

Training Train all the GMMs on their associated audio data with the Expectation Maximization (EM) algorithm ([Bishop, 1995], page 65).

Segmenting Run a Viterbi to re-segment the data.

Merging Based on the the ΔBIC score (see equation 3.1) [Ajmera, 2003], a variation of the Bayesian Information Criterion (BIC) score, decide if there are clusters to merge. Repeat the training-segmenting-merging routine until there are no more clusters to merge.

BIC imposes a trade-off between model quality and model complexity. A problem of the BIC is that it depends on a parameter λ that needs to be tuned. This tunable parameter is eliminated in the ΔBIC score. Basically it is possible to get rid of the λ if models with the same number of free parameters are compared. In the presented approach, two models (GMMs) containing $M1$ and $M2$ Gaussians are possibly merged into a GMM made up of $M1 + M2$ Gaussians, thus the number of free parameters are the same before and after merging. At every merging step, for every pair of clusters, a merged model is trained and ΔBIC is calculated. A pair should be merged if ΔBIC is larger than zero. If there are more cluster pairs that have a ΔBIC that is larger than zero, the pair with the highest one is merged. ΔBIC can be calculated as follows:

$$\Delta BIC = \log p(D|\theta) - (\log p(D_a|\theta_a) + \log p(D_b|\theta_b)) \tag{3.1}$$

where D_a is the data assigned to cluster a, D_b is the data assigned to cluster b and D contains $D_a \cup D_b$. θ, θ_a and θ_b represent the parameters of the corresponding PDFs.
It is shown in [Huang et al., 2007] that finding the best pair to merge and merging takes 62% of the runtime of the system. The so-called fast-match component presented in [Huang et al., 2007] reduces the hypothesis space of the expensive model selection and speeds up the described speaker diarization system by 41% without affecting the accuracy. In the current implementaion of the ICSI Speaker Diarization engine that uses only MFCCs, this fast-match component is used.

Chapter 4

Problem statement

I started writing on this thesis because the performance of the ICSI Speaker Diarization engine (see Chapter 3) dropped off dramatically for shorter meetings. The intuition of my colleagues was that this had to do with the last set of free parameters that still had to be tuned manually. Therefore a performance analysis of the ICSI Speaker Diarization engine on short meetings is presented in Section 4.1. The behavior under variation of parameters is then shown in Section 4.2 and in the last section of this chapter, the goals of this master thesis are concluded.

4.1 Behavior of the engine for shorter meetings

In order to simulate the behavior of the diarization engine for short meetings, the meetings of the NIST RT-06 development set (see appendix A for a list of the meetings that form the used data-sets) are split into smaller pieces of different durations. For the first experiments, the meetings are cut into parts of 10, 25, 50, 75, 100, 200, 300, 400 and 500 seconds segments (the total duration of the meetings of this data-set is between 600 and 700 seconds). The results are shown in Figure 4.1. The figure presents the error rates of the speech/non-speech detection, the Speaker Error Rate as well as the total Diarization Error Rate. The Diarization Error Rate is the primary metric for the NIST RT evaluations of the Speaker Diarization task and is defined as follows:

$$DER = \frac{\sum_{all\ \ segments}\{dur(seg) \cdot (max(N_{ref}(seg), N_{sys}(seg)) - N_{correct}(seg))\}}{\sum_{all\ \ segments} dur(seg) \cdot N_{ref}(seg)}$$

where the speech data file is divided into contiguous segments at all speaker change points and where, for each segment seg:
$dur(seg)$ = the duration of seg
$N_{ref}(seg)$ = the # of reference speakers speaking in seg
$N_{sys}(seg)$ = the # of system speakers speaking in seg
$N_{correct}(seg)$ = the # of reference speakers speaking in seg for whom their matching

(mapped) system speakers are also speaking in *seg*.
The speech/non-speech error is almost constant even for shorter segments, but the Speaker Error is clearly growing as the duration of the meeting segments becomes shorter. At first, it seemed to be surprising that the Speaker Error gets smaller for segments of less than 100 seconds and drastically drops down for segments of 10 and 25 seconds. It was suspected that there are only very few speakers in the shorter segments, but Figure 4.2 shows the speaker distribution of the ground truth files. Every subplot shows how many percent of the segments (y-axis) contain how many speakers (x-axis). It can be seen, that the speaker distribution changes, but the peak of the distribution is still at $x = 2$ speakers and almost 80% of the 10 seconds segments contain more than one speaker ($y \approx 0.2$ for $x = 1$). Analyzing in more detail the results shows that for the shortest split duration (10 seconds) the system finds many times only one speaker and gets a very good evaluation. After having listened to some of the meetings, it became clear, that often one speaker is speaking and another one is just nodding or confirming what was said (what actually counts as another speaker participating in the meeting). In Figure 4.3, the speaker distribution for the segments of 10 seconds is shown when speech segments of less than one seconds are filtered out. It can be seen that the peak is now at $x = 1$ speaker and in almost 50% of the segments, there is no need of Speaker Diarization. The good evaluation results can be explained with the time-based evaluation of the NIST DER.
Further, we may think of the simplest Speaker Diarization engine that assigns one cluster to the whole segment to diarize. In Figure 4.4, the ICSI Speaker Diarization engine is compared to an engine with that *one-cluster-assignment* strategy. The two performance curves intersect for a split duration slightly above 100 seconds. This underlines the poor performance of the ICSI Speaker Diarization engine for such short segments.
These observations motivate the need of improvement of the ICSI Speaker Diarization engine for short meetings and also justifies a lower bound of 100 seconds for the segment durations.

4.2 Sensitive Parameters

The core of the current ICSI Speaker Diarization engine is presented in [Ajmera, 2003]. The engine is described to have so called hyper-parameters which are claimed to be insensitive and there should be no need to tune them. The engine takes the number of initial clusters K, the number of Gaussians per initial cluster M and the minimal duration of a speech segment MD as input (see Section 3.3). Moreover, there may be some hidden parameters in the engine such as the number of iterations during the GMM training. To get a first impression of the parameters,

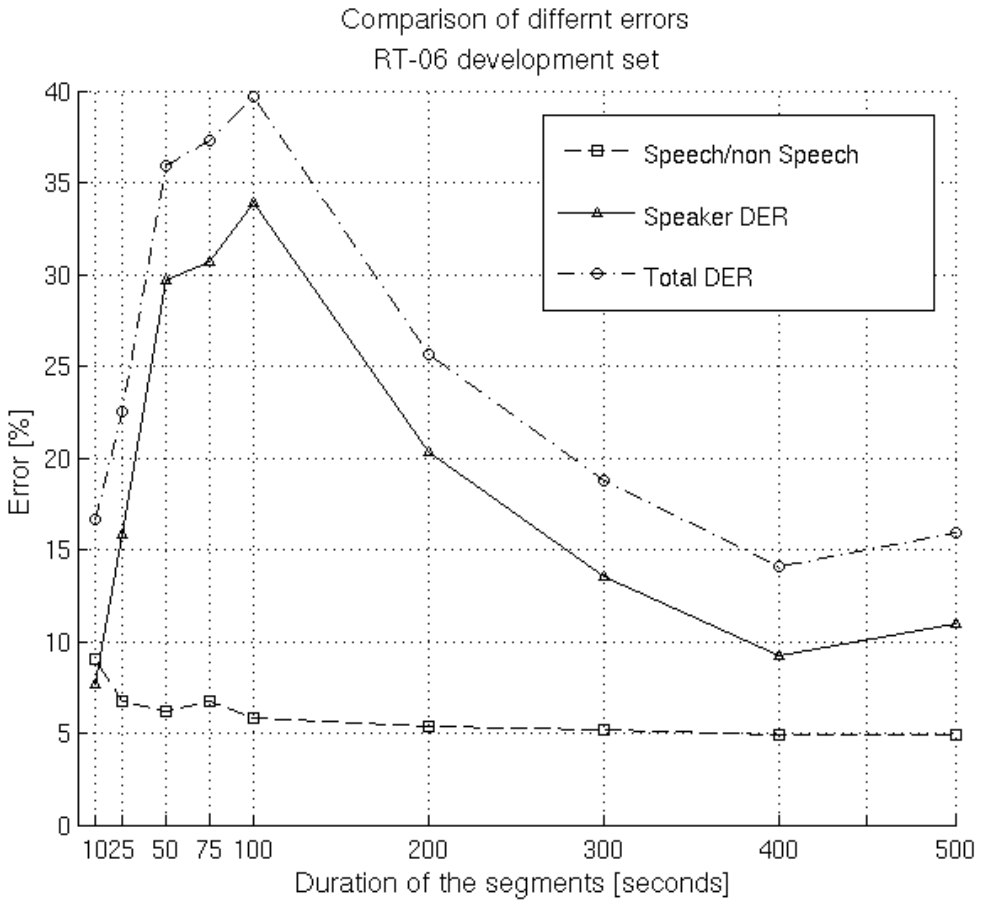


Figure 4.1: Evaluation of the ICSI Speaker Diarization engine performance for segments of different durations. The x-axis represents the duration of the segments in seconds and the y-axis the different errors in percent. The speech/non-speech error, the Speaker Error and the total Diarization Error Rate (DER) (see Section 4.1) is shown. If the meeting segments become shorter, the speech/non-speech error remains almost constant, whereas the total DER is growing until the segments are made of 100 seconds and if the segments are even shorter, the total DER is falling down again.

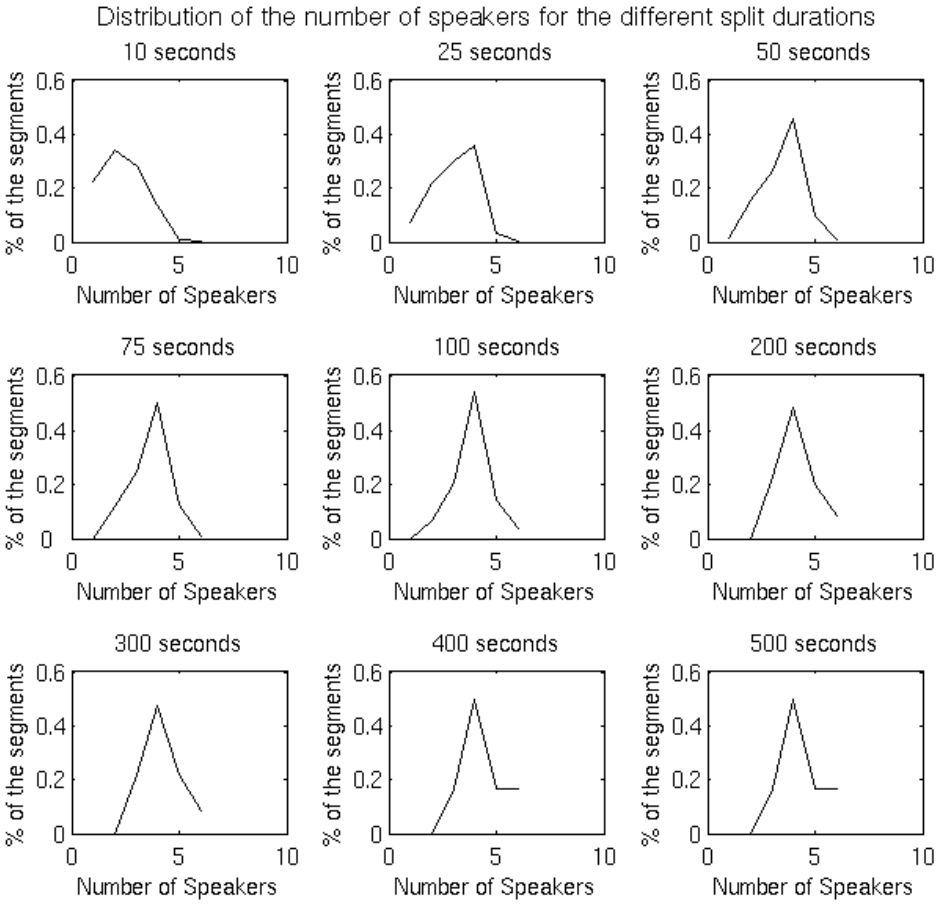


Figure 4.2: Speaker Distribution for the different split durations. Every subplot contains all data of the RT-06 development set. The y-axis represents the percentage of the segments that contain a certain number of speakers mentioned on the x-axis. For the longest segments, the peak is at $x = 4$ speakers, for the shortest segment duration (10 seconds) the peak is at $x = 2$ and almost 80% of the 10-second segments contain more than one speaker.

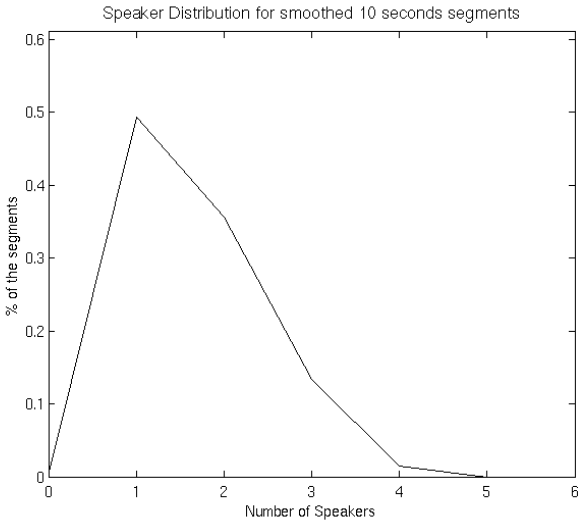


Figure 4.3: Smoothed Speaker Distribution for 10 seconds segments. On the x-axis are the number of speakers and on the y-axis the percentage of segments that contain a certain number of speakers. Speech segments that are shorter than one second are smoothed out. In that case almost 50% of the segments contain only one speaker and in that case there is no need of Speaker Diarization.

some preliminary experiments should show if the parameters are sensitive or not. For that purpose, the parameters K and M are varied around their *default* values which are $K = 16$ and $M = 5$. During one set of experiments, the number of initial clusters K is varied while the number of Gaussians per initial cluster M is hold constant and during another set of experiments K is hold constant and M is varied. The results are listed in table 4.1 and show that the parameters are in fact very sensitive to small variations.

Number of initial cluster K ($M = 5$)	14	15	16	17	18
Speaker Error	9.7%	7.6%	6.9%	6.0%	6.3%
Gaussians per initial cluster M ($K = 16$)	3	4	5	6	7
Speaker Error	8.9%	8.0%	6.9%	9.8%	8.9%

Table 4.1: Results of some preliminary experiments to show the sensitivity of the parameters. During one set of experiments K is varied and M is set to five, during another set of experiments M is varied and K is set to 16. Even small variations of the parameter values may affect the Speaker Error.

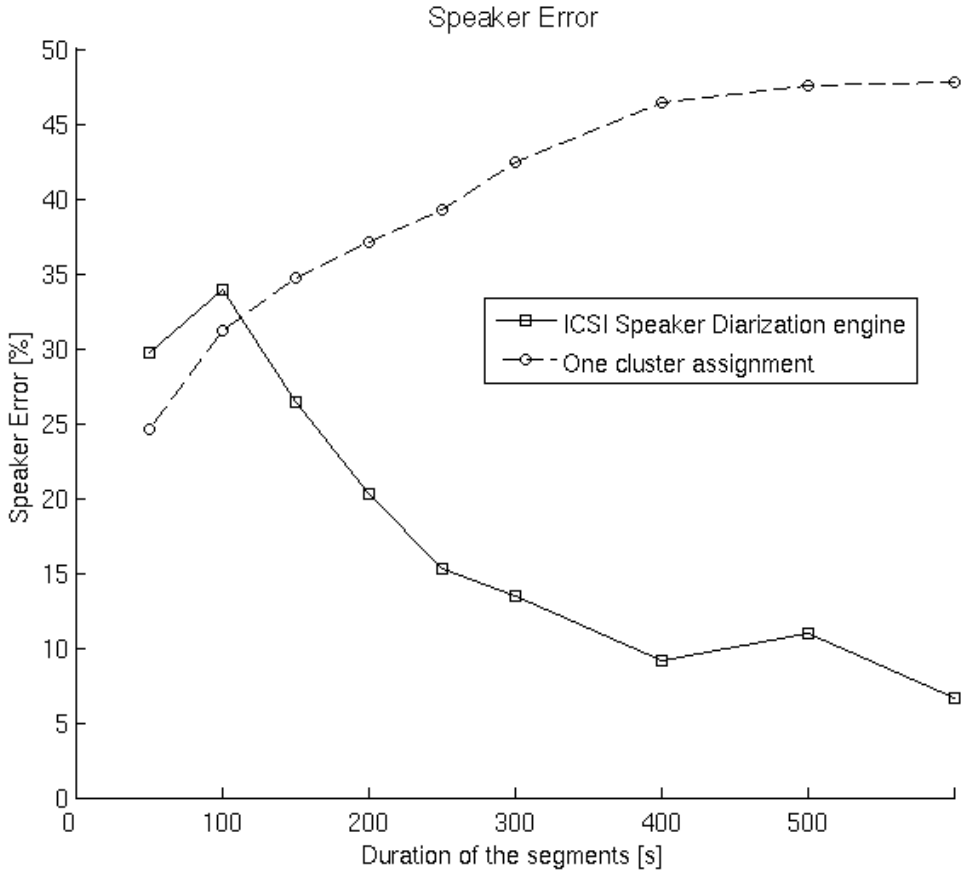


Figure 4.4: Comparison of the performances of the ICSI Speaker Diarization engine and the simplest assignment strategy which is assigning one cluster to the whole segment. On the x-axis, the duration of the segments in seconds is shown and the y-axis represents the Speaker Error in percent. The simplest diarization engine with a *one-cluster-assignment* strategy performs better than the ICSI engine for segments of 100 seconds and less, what underlines the very poor performance of the ICSI Diarization engine for short meetings.

4.3 The Problem

The main goal of the work is to flatten the performance curve of the ICSI speaker diarization engine shown in Figure 4.1 and to make the engine perform better on shorter meetings. The ideal case is that the performance of the engine does not at all depend on the duration of the meeting. This would also be an important step towards on-line diarization. Furthermore as seen in Section 4.2, the engine is very sensitive to minor changes in the parameters. Therefore it is important to reduce their number to increase the robustness of the ICSI Speaker Diarization system, since it is hard to predict how the system performs on unseen data otherwise.

Chapter 5

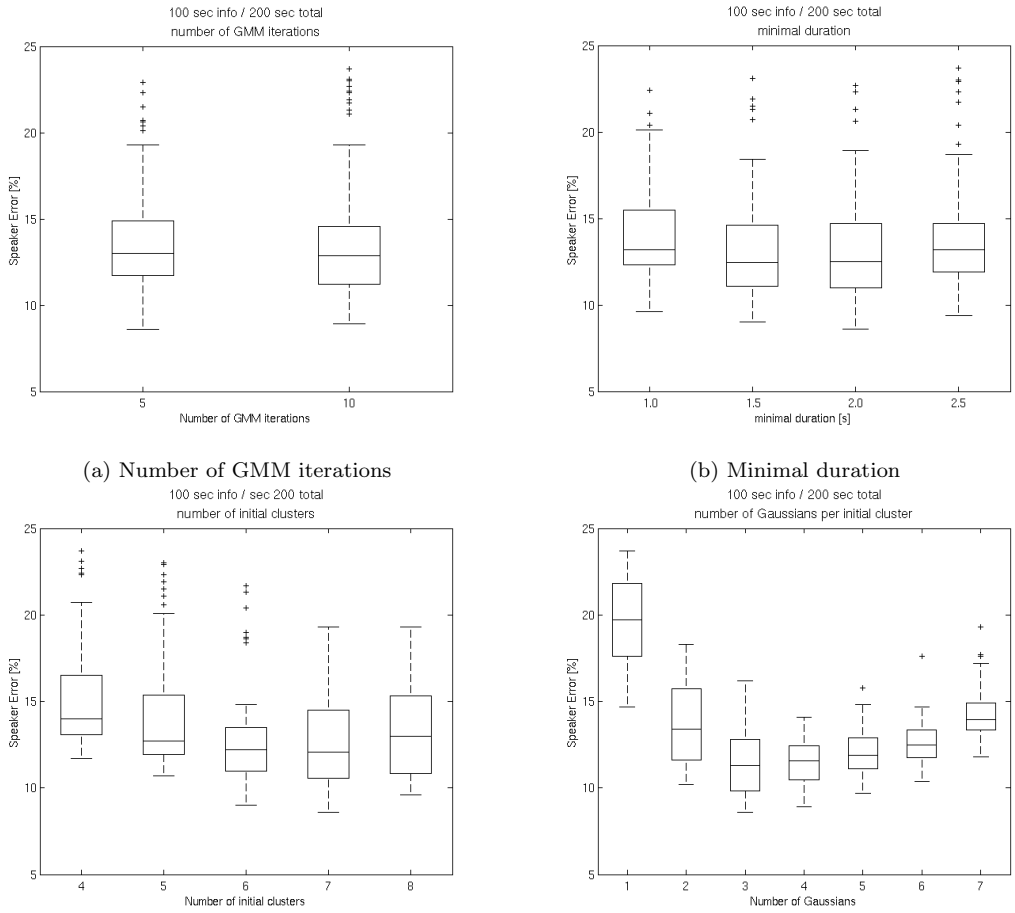
Short meetings

This chapter describes the influence of the parameters of the ICSI Speaker Diarization engine if the engine is dealing with shorter meetings. There are four parameters to set and it is intuitively hard to say how much influence they have and how the engine behaves when the meetings are shorter and the parameters are changed. In Section 5.1 the results of an exhaustive search will show which parameters are important and if it is possible to improve the result for shorter meetings by tuning the parameters. In Section 5.3, relations between parameters, that have a high correlation among the best performing configurations of the exhaustive search, are presented and it is shown how these relations can be exploited by building models with the help of a linear regression. The test of these models is presented in 5.4 and the chapter concludes in Section 5.5 where some results are further analyzed by using a Graphical User Interface.

5.1 Exhaustive search

5.1.1 Experimental setup

For the exhaustive search, the following parameters are taken into consideration: the minimal duration of a speaker segment (model constraint), the number of Gaussians per initial cluster, the number of initial clusters and the number of GMM iterations. The target of the exhaustive search is to understand which parameters are important and meeting duration dependent. At a first time, experiments are run on 100-second segments that are duplicated. Every segment contains only 100 seconds of information, but these 100 seconds are duplicated and concatenated. This special configuration is used because it promises good results (for more details, see Section 5.4.3). The results are plotted in form of boxplots in Figure 5.1. In each subfigure (a, b, c and d) the same data is presented, just sorted by a different parameter. It seems that the number of GMM iterations and the minimal duration does not have as much influence on the results as the number of Gaussians per initial cluster



(a) Number of GMM iterations

(b) Minimal duration

(c) Number of initial clusters

(d) Number of Gaussians per initial cluster

Figure 5.1: Boxplots of the performance of the ICSI Speaker Diarization engine during the exhaustive search experiments for 100-second segments described in Section 5.1.1. All subplots contain all the results but are sorted by a different parameter which is listed on the x-axis. On the y-axis, the Speaker Error in percent is shown. It can be seen that the variations of the number of initial clusters and the number of Gaussians per initial cluster have most influence on the Speaker Error.

and the number of initial clusters. Based on these results, more experiments on different segment durations are run. Therefore, the whole audio track is split into segments of x seconds where $x \in \{100, 150, 200, 250, 300\}$. The parameters are varied around the values that were chosen for the RT-06 development set: number of initial clusters $K = 16$, number of Gaussians per initial cluster $M = 5$ and the minimal duration of a speech segment $MD = 2.5$. Based on the results for 100-second segments presented earlier in this section, the minimal duration of a speech segment is only set to 1.5 seconds, 2.0 seconds and 2.5 seconds. The variation of that paramater shows no significant influence on the meetings of that data-set and is therefore discarded for further tuning experiments. On the other hand, the number of initial clusters and the number of Gaussians per initial clusters are varied on larger intervals depending on temporary experiment evaluations with the goal to localize a minimum for both parameters in the neighborhood of the $default$ values. The computational cost to vary the number of GMM iterations in a large range is too high compared to the expected improvement.

5.1.2 Results

For every segment duration another configuration achieves the best result in terms of minimizing the Speaker Error. The best performing configurations can be seen in Table 5.1. It is interesting that just by tuning the engine, even for very short

Duration of the segments	Number of initial clusters	Number of Gaussians per initial cluster	Speaker Error
100s	13	2	11.3 %
150s	16	2	10.0 %
200s	13	3	9.1 %
250s	7	4	8.4 %
300s	14	4	7.6 %

Table 5.1: Best performing configurations for every segment duration with the corresponding Speaker Error and the parameter settings for the number of initial clusters and the number of Gaussians per initial cluster. The ICSI Speaker Diarization engine performs much better on short segments with these parameter settings than with the parameter settings that were chosen by the manual tuning to longer segments (see Section 5.1.2 and Figure 5.2).

segment durations, the ICSI Speaker Diarization engine performs relatively well. This is shown in Figure 5.2 where three different curves are displayed. The version of

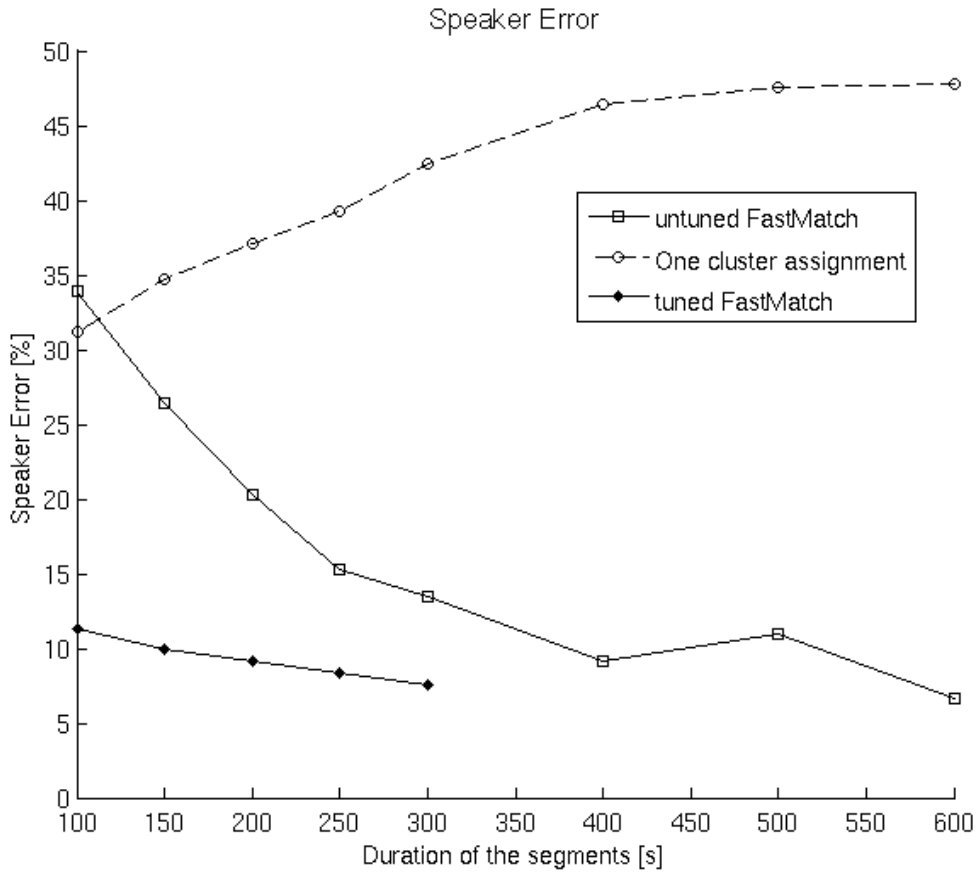


Figure 5.2: Comparison of the tuned engine that uses the parameters that performed best during the exhaustive search experiments (see Table 5.1) to the untuned engine (number of initial clusters is set to 16 and five Gaussians per initial cluster are used) and the one-cluster-assignment strategy (see Section 4.1). Every point represents the average Speaker Error in percent (y-axis) over the 12 meetings from the RT-06 development set split into segments of a certain duration (x-axis). The curve of the tuned engine is much flatter, thus tuning increases the performance of the engine for shorter segment durations.

the ICSI Speaker Diarization system that is used contains the fast-match component that is presented in [Huang et al., 2007] and is therefore labeled with *FastMatch*. The untuned version uses the same parameter values ($K = 16$, $M = 5$) for every segment length and the tuned version uses the values mentioned in Table 5.1. There are no results for the tuned version for segment durations of more than 300 seconds because these segment durations were not part of exhaustive search experiments. The third curve displays the Speaker Error if the *one-cluster-assignment* strategy (see Section 4.1) is applied. Every data point respresents the average Speaker Error over all the meetings in the RT-06 development set split into segments of a certain duration. The performance of the untuned engine clearly drops off for shorter segments whereas the curve of the tuned engine is much flatter. Therefore it is concluded that tuning the parameters of the engine improves the performance on shorter segments.

5.2 Interpretation and visualization of the results

The previous section has shown that tuning the parameters increases the performance, but this tuning can not be done manually for every segment duration and needs to be done automatically. In order to better understand and interpret the results, one part of the analysis consists of visualizing the results with the help of Weka [Weka, 2008]. The visualization should show dependencies for the best performing configurations between the segment duration and the most promising tunable parameters that are the number of Gaussians per initial cluster M and the number of initial clusters K. Inspired by [Leeuwen and Konečný, 2008] who are specifying the amount of seconds available to train one single Gaussian, another interesting parameter, the seconds per Gaussian is defined to be the seconds of speech available divided by the total number of Gaussians, used during the clustering process: $secpergauss = \frac{\text{speech duration in seconds}}{M \cdot K}$. This parameter is a combination of the two tunable parameters (the number of initial clusters K and the number of Gaussians per initial cluster M) and, in fact, it can be seen in Figure 5.3, that the parameter seems to have a minimum that is very similar for different segment durations and one can clearly recognize a curve.
To concentrate on only one tunable parameter at the moment, based on the boxplots shown in Figure 5.4 it is decided to fix the number of Gaussians per initial cluster M to four and to concentrate on the analysis and estimation of the number of initial clusters K. The number of Gaussians per initial clusters is fixed to four even if the mean Speaker Error is not the lowest because for $M = 4$ the lowest overall minimum Speaker Error is obtained and that boxplot shows the least extrem outliers.

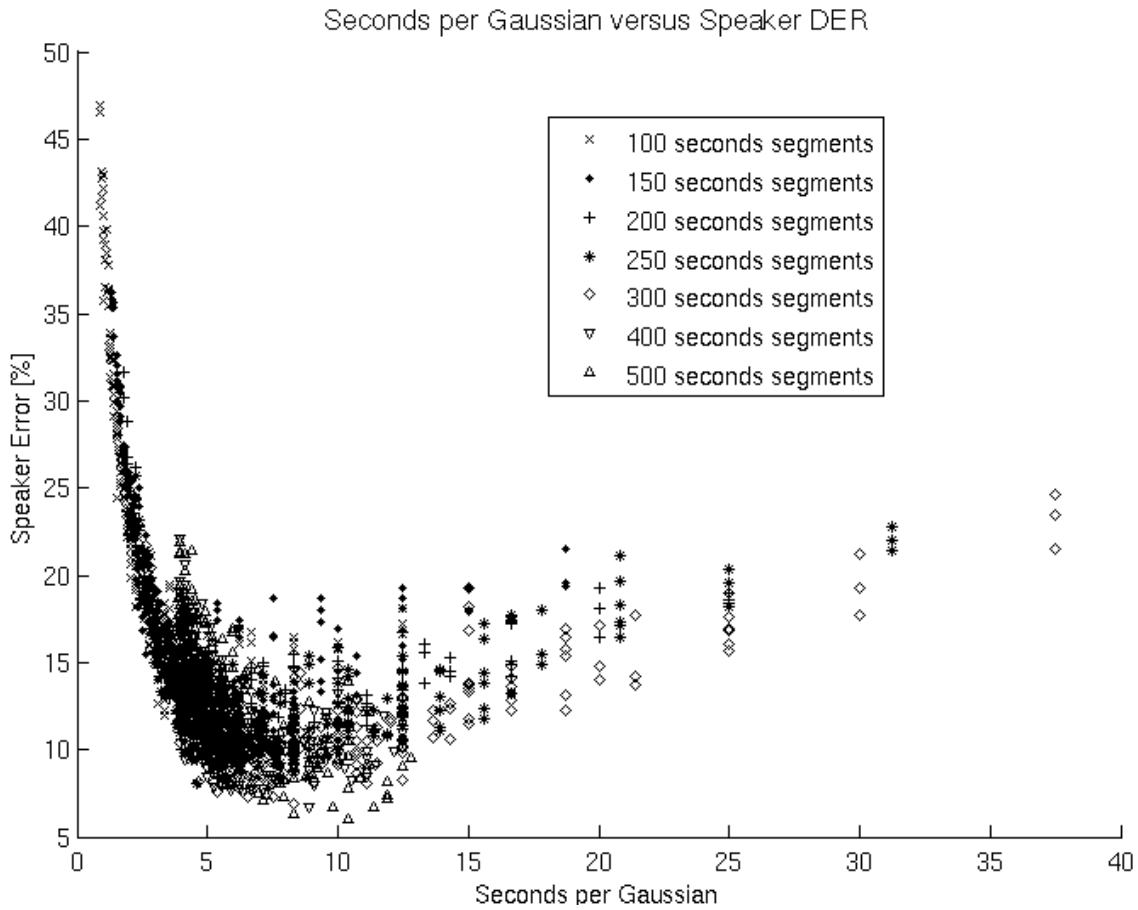


Figure 5.3: The Speaker error in percent (y-axis) versus the seconds per Gaussian (x-axis, see Section 5.2). Every data point corresponds to the average Speaker Error of 12 meetings for one particular configuration tested during the exhaustive search. Configurations for all tested segment durations are shown in the same plot and one can clearly recognize some kind of curve (or a combination of curves, one curve for every segment duration). These curves seem to have a minimum that is similar for different segment durations.

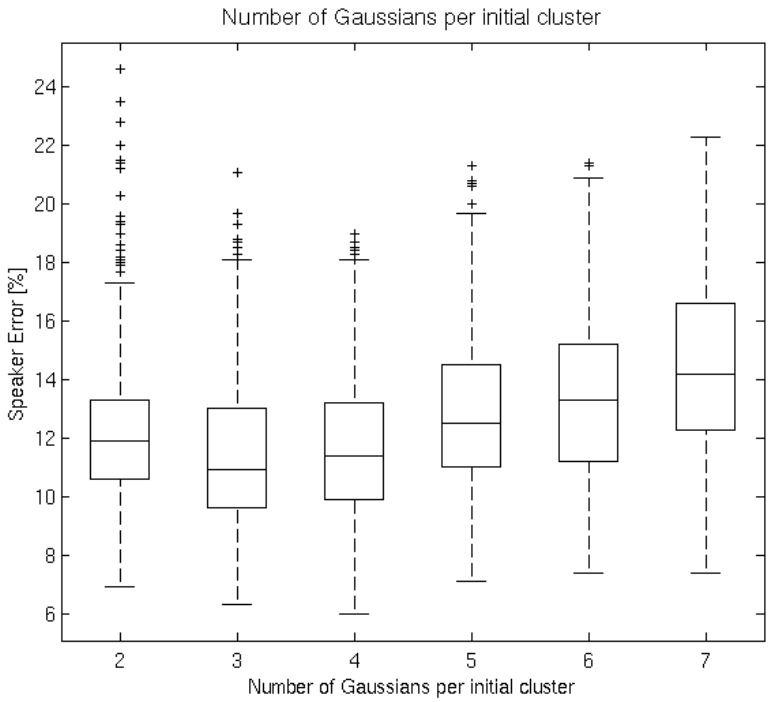
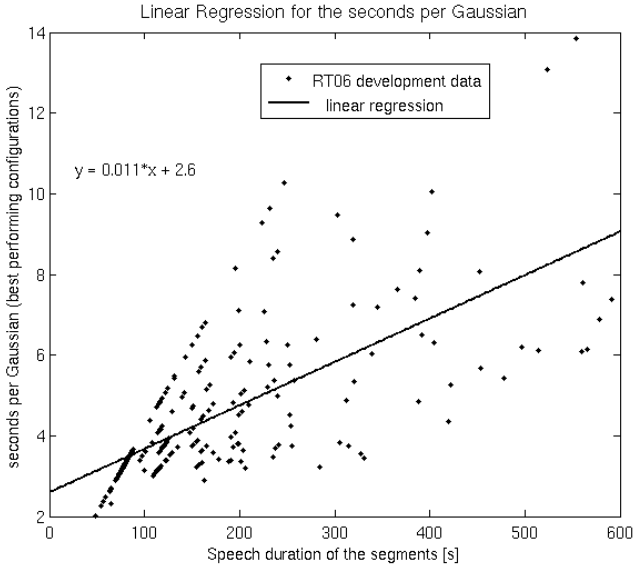


Figure 5.4: Boxplots representing the experiments of the exhaustive search for different segment durations (100, 150, 200, 250, 300 seconds) sorted by different number of Gaussians per initial cluster (x-axis). The y-axis shows the average Speaker Error in percent. Four Gaussians per initial cluster show a very good overall performance (least extrem outliers), a very good mean (only slightly worse than the mean of three Gaussians per initial cluster) and the lowest Speaker Error was obtained with four Gaussians per initial cluster.

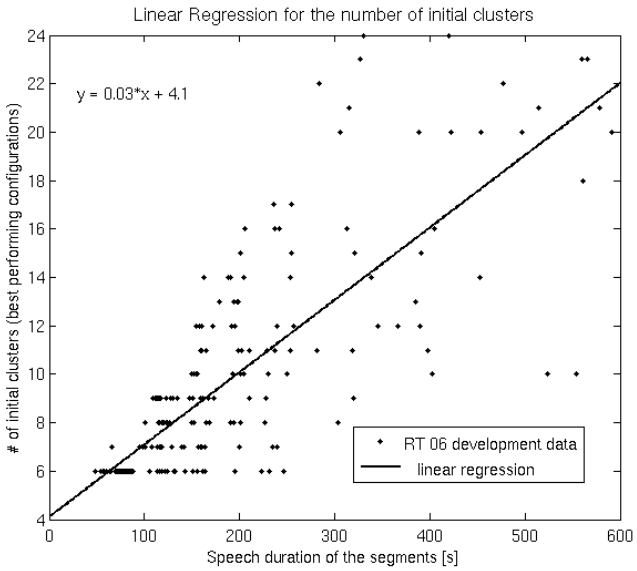
5.3 Linear Regression models

The last sections have shown that two tunable parameters are most significant, namely the number of initial clusters (K) and the number of Gaussians per initial cluster (M). In the previous section, another parameter, the seconds per Gaussian (that depends on the speech duration, M and K) was introduced. To concentrate on one parameter at a time, the number of Gaussians per initial cluster was fixed to four (see Section 5.2). Among all tested configurations, the best performing ones per segment duration x where $x \in \{100, 150, 200, 250, 300\}$ is searched. Visualizing the speech duration of every processed segment versus the corresponding number of initial clusters respectively the resulting seconds per Gaussian for the best configurations justified the calculation of the correlations between these relations in Matlab and it turned out that the correlations are indeed relatively high. The correlation values are 0.6756 for the relation between the speech duration of the segments and the seconds per Gaussian and 0.776 for the relation between the speech duration of the segments and the number of initial clusters (see Figure 5.5). Both relations lead to the tunable parameter K (number of initial clusters), because $K = \frac{speech\ duration\ in\ seconds}{seconds\ per\ Gaussian \cdot number\ of\ Gaussians}$. The exhaustive search has shown that tuning the parameters of the ICSI Diarization engine can really improve the performance (see Figure 5.2), but the tuning should be done automatically, without any prior training to some specific meeting data. The speech duration is a known parameter after the speech/non-speech detection and there is a relatively high correlation between the speech duration and the number of initial clusters respectively the seconds per Gaussian for the best performing configurations. Therefore, the linear regressions, calculated in Matlab and represented in Figure 5.5, are used to build a linear model for a parameter choice that depends on the speech duration of a segment.

The performance of model 1 (seconds per Gaussian) and model 2 (number of initial clusters) on the RT-06 development data can be seen in Figure 5.6 and the exact Speaker Errors are printed in Table 5.2. In Figure 5.6, every datapoint correspond to the average results of all RT-06 meetings cut into segment of duration x, where $x \in \{100, 150, 200, 250, 300, 400, 500, tot\}$, tot means processing the whole meeting as one segment (duration is meeting dependent, but larger than 500 seconds). The x-coordinate in the plot corresponds to the average effective speech duration. Both models perform very similar for segment durations up to 300 seconds. For segments of 400 seconds, 500 seconds and complete meetings, model 2 performs very well and better than model 1 and the untuned FastMatch engine (ICSI baseline). However, only for segment durations of 400 seconds, model 1 did significantly worse than the ICSI baseline system. For the complete meetings configuration, model 1 and the ICSI baseline perform not significantly different and for a segment duration of 500 seconds, model1 did better than the ICSI baseline. The models are trained on the



(a) Seconds per Gaussian



(b) Number of initial clusters

Figure 5.5: The speech duration of the segments (x-axis) versus the seconds per Gaussian (a) respectively the number of initial clusters (b) for the best performing configurations of every segment duration. Every datapoint correspond to one processed segment. The relations show a relatively high correlation (see Section 5.3) and the linear regression (calculated with Matlab) is also shown in the plots.

RT-06 development data and it is expectable that they perform better on short segment durations than a static system that was tuned to the complete meeting length. In the next section, the models are tested on other data-sets.

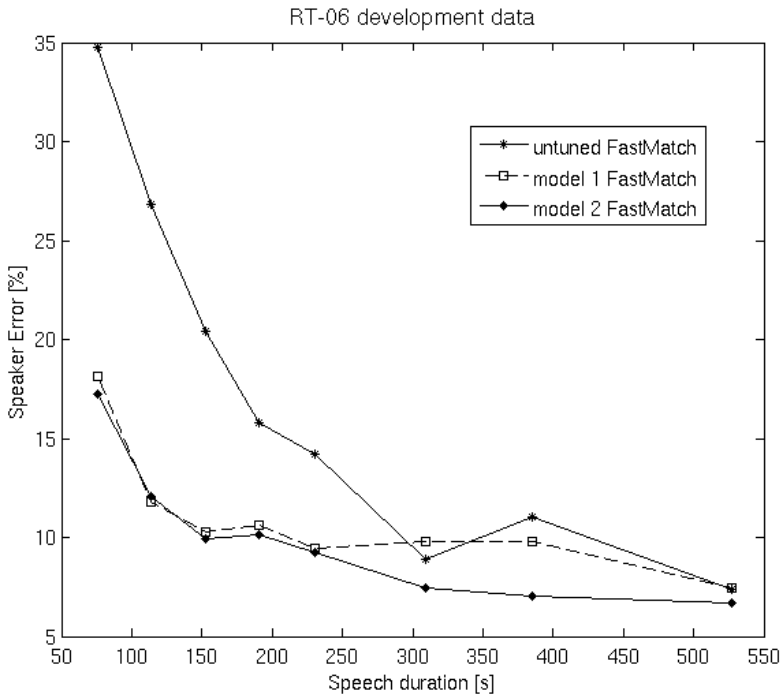


Figure 5.6: Performance of the linear regression models on the RT-06 development data. On the x-axis, the speech duration in seconds is displayed (every point corresponds to the average speech duration over the RT-06 development set split into segments of a particular duration) and on the y-axis the average Speaker Error is shown. Model1 is the linear regression model based on the seconds per Gaussian and model2 is the one based on the number of initial clusters. The performance of both models is very similar and significantly better than the untuned FastMatch engine for segment durations of less than 300 seconds. For larger segments model2 performs better than model1.

5.4 Testing the regression models

5.4.1 A set of chosen AMI meetings

To test the models, a set of 12 AMI meetings (all containing 4 speakers) is chosen (the meetings are listed in appendix A, where the individual Speaker Error Rates

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	7.40 %	-
	model1	7.43 %	+0.4 %
	model2	6.67 %	-9.9 %
500	FastMatch (baseline)	11.02 %	-
	model1	9.80 %	-11.1 %
	model2	7.02 %	-36.3 %
400	FastMatch (baseline)	8.93 %	-
	model1	9.81 %	+9.9 %
	model2	7.43 %	-16.8 %
300	FastMatch (baseline)	14.18 %	-
	model1	9.47 %	-33.2 %
	model2	9.22 %	-35.0 %
250	FastMatch (baseline)	15.80 %	-
	model1	10.63 %	-32.7 %
	model2	10.11 %	-36.0 %
200	FastMatch (baseline)	20.38 %	-
	model1	10.31 %	-49.4 %
	model2	9.94 %	-51.2 %
150	FastMatch (baseline)	26.81 %	-
	model1	11.83 %	-55.9 %
	model2	12.04 %	-55.1 %
100	FastMatch (baseline)	34.74 %	-
	model1	18.15 %	-47.8 %
	model2	17.23 %	-50.4 %

Table 5.2: Comparison of the performance of the linear regression models model1 (based on the seconds per Gaussian) and model2 (based on the number of initial clusters) to the ICSI baseline system for RT-06 development set. Only the average Speaker Error is presented. The speech/non-speech error is basically the same for the ICSI baseline and the linear regression model approaches and therefore not listed in the table. The relative change shows that the linear regression models perform better than the ICSI baseline system for most configurations.

per meeting are listed). The results of the experiments can be seen in Figure 5.7 and the exact Speaker Errors are listed in Table 5.3.
The 12 meetings are longer than the meetings of the RT-06 development set. This

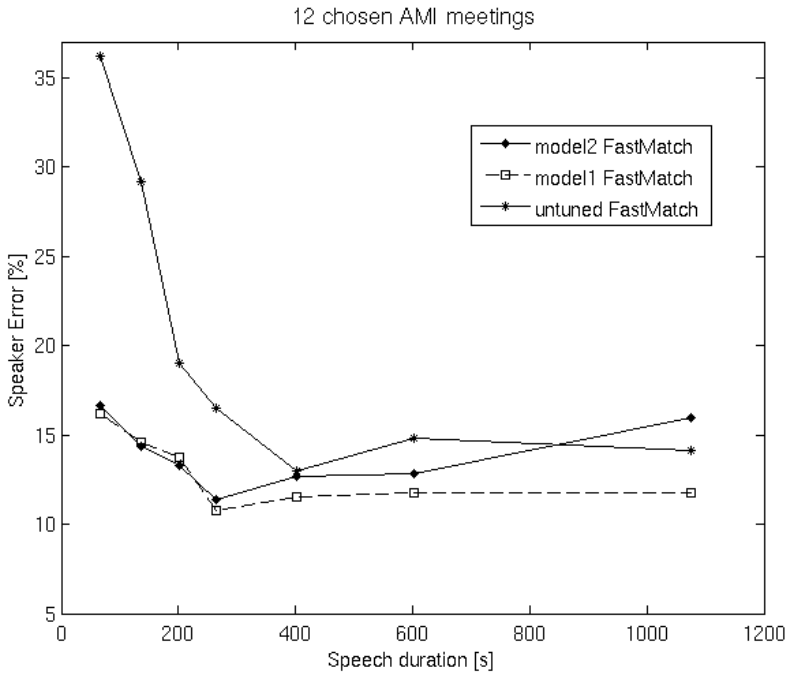


Figure 5.7: Performance of the linear regression models on 12 chosen AMI meetings. On the x-axis, speech duration in seconds is displayed (every point corresponds to the average speech duration of the 12 meetings split into segments of a particular duration x where $x \in \{100, 200, 300, 400, 500, 900, tot\}$) and on the y-axis the average Speaker Error is shown. Model1 is the linear regression model based on the seconds per Gaussian and model2 is the one based on the number of initial clusters (see Section 5.3). The performance of model1 is very impressive because it performs significantly better than the ICSI baseline system for all segment durations, whereas model2 performs worse than the ICSI baseline system if the complete meetings are processed.

time the meetings are split into segments of duration x, where $x \in \{100, 200, 300, 400, 500, 900, tot\}$ (tot stands for processing the whole meetings). Again the x-coordinate corresponds to the average effective speech duration for a certain segment duration. Model 1 performed very well whereas model 2 performed poorly in the region of longer durations. This could be explained by the fact, that model 2 may be overfitted to the RT-06 development set and the durations of that meetings. Further, it can be seen, that model 1 clearly performs better than the ICSI baseline system on

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	14.38 %	-
	model1	11.77 %	-18.2 %
	model2	15.99 %	+11.2 %
900	FastMatch (baseline)	14.85 %	-
	model1	11.73 %	-21.0 %
	model2	12.79 %	-13.9 %
500	FastMatch (baseline)	12.98 %	-
	model1	11.54 %	-11.1 %
	model2	12.66 %	-2.5 %
400	FastMatch (baseline)	16.53 %	-
	model1	10.76 %	-34.9 %
	model2	11.41 %	-31.0 %
300	FastMatch (baseline)	19.04 %	-
	model1	13.71 %	-28.0 %
	model2	13.27 %	-30.3 %
200	FastMatch (baseline)	29.14 %	-
	model1	14.62 %	-49.8 %
	model2	14.38 %	-50.7 %
100	FastMatch (baseline)	36.22 %	-
	model1	16.17 %	-55.4 %
	model2	16.68 %	-54.0 %

Table 5.3: Comparison of the performance of the linear regression models (see Section 5.3) model1 (based on the seconds per Gaussian) and model2 (based on the number of initial clusters) to the ICSI baseline system (FastMatch) for 12 chosen AMI meetings and different split durations. Only the Speaker Error is presented. The speech/non-speech error is basically the same for all approaches and therefore not listed in the table. The relative change shows that the model1 performs better than the ICSI baseline system for all durations whereas model2 performs worse for complete meetings.

the average of the 12 chosen AMI meetings, not only for short segment durations but also when processing the complete meetings. Because of the impressive performance of the linear regression model based on the seconds per Gaussian it is decided to test that model on more data sets and it is claimed that this model is generalizable. On the other hand, because of the relatively bad performance of the model based on the number of initial cluster, it is not tested on other data-sets but discarded.

5.4.2 NIST RT-06 and NIST RT-07 evaluation sets

The better performing linear regression model (model1, based on the seconds per Gaussians) is also tested on the evaluation sets of the NIST RT-06 and RT-07 (see appendix A for a list of the meetings in every data set). A comparison between the untuned engine ($K = 16$ and $M = 5$ for all segment durations) and the parameter choice based on the linear regression model can be seen in Figures 5.8 and 5.9 and the exact average Speaker Errors are listed in Table 5.4 to 5.7.
The Multiple Distant Microphone (MDM) and Single Distant Microphone (SDM)

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	16.68 %	-
	model1	15.16 %	-9.1 %
500	FastMatch (baseline)	14.70 %	-
	model1	13.27 %	-9.7 %
300	FastMatch (baseline)	19.82 %	-
	model1	13.32 %	-32.8 %
100	FastMatch (baseline)	40.23 %	-
	model1	18.73 %	-53.4 %

Table 5.4: Comparison of the performance of the linear regression model based on the seconds per Gaussian to the ICSI baseline system (FastMatch) for the 9 meetings of the RT-06 evaluation set (MDM recordings) and different split durations. Only the Speaker Error is presented. The speech/non-speech error is basically the same for both approaches and therefore not listed in the table.

recordings are processed and evaluated. The different channels of the MDM recording permit to perform beamforming to create one enhanced channel. This channel is then used to perform Speaker Diarization and explains the better performance of the MDM case. For the NIST RT-06 evaluation set, the comparison of the MDM and the SDM condition is not relevant because based on naming convention problems one SDM recording meeting (TNO_20041103-1130) is excluded from the experiments. *Untuned engine* denotes the manually tuned engine with the parameter settings that

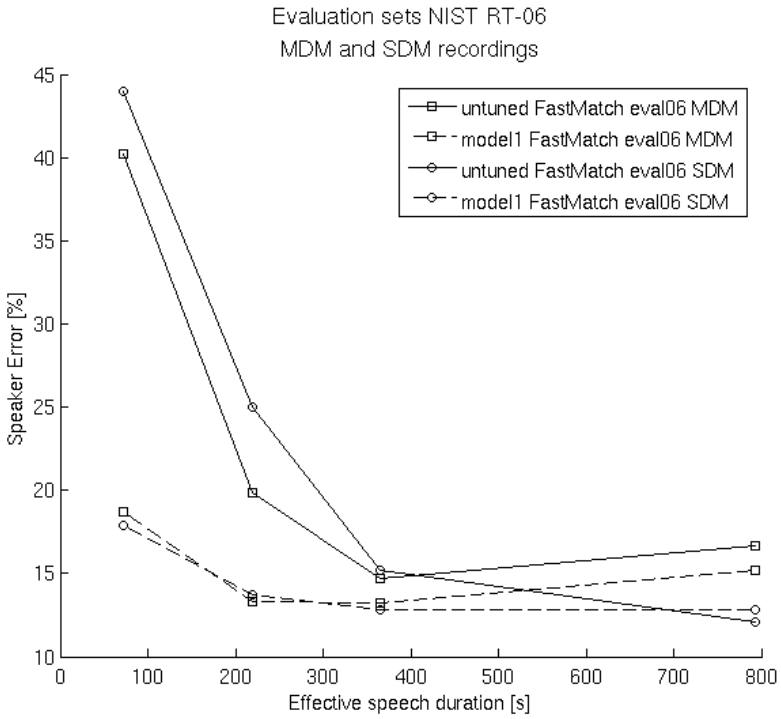


Figure 5.8: Performance of the linear regression model on the evaluation set of RT-06. On the x-axis, the average speech duration over the nine (respectively eight for the SDM case, see Section 5.4.2) meetings split into segment durations of $x \in \{100, 300, 500, tot\}$ is shown and the y-axis represents the average Speaker Errors in percent. The Multiple Distant Microphone (MDM) and Single Distant Microphone (SDM) recordings performances are displayed in the same plot. The linear regression model performs better than the baseline system on all configurations, except the complete meeting case of the SDM recordings. The comparison of the SDM and MDM recordings is not relevant because the sets do not contain the same number of meetings.

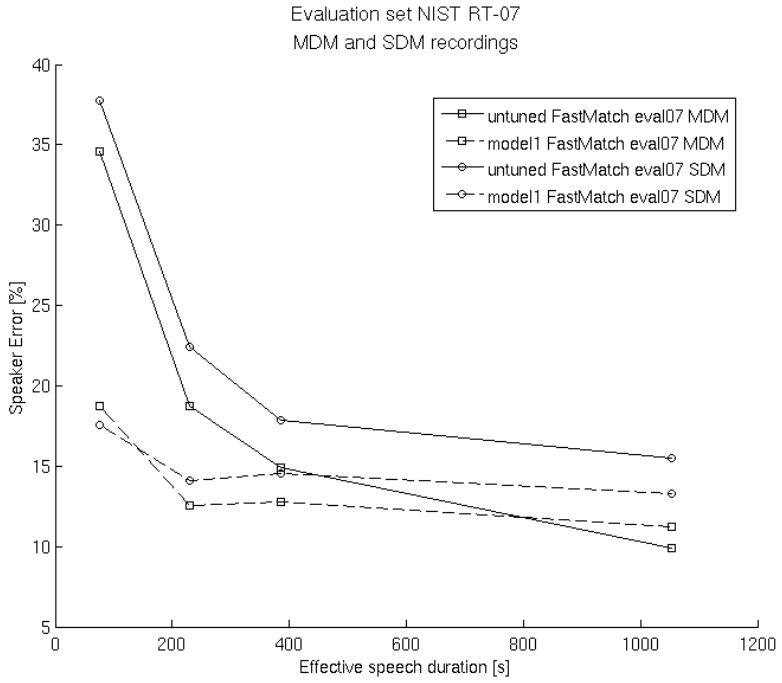


Figure 5.9: Performance of the linear regression model on the evaluation set of RT-07. On the x-axis, the average speech duration over the eight meetings split into segment durations of $x \in \{100, 300, 500, tot\}$ is shown and the y-axis represents the average Speaker Errors in percent. The Multiple Distant Microphone (MDM) and Single Distant Microphone (SDM) recordings performances are displayed in the same plot. The linear regression model performs better except for the complete meeting case of the MDM recordings. Further, the results of the linear regression model for the SDM recordings are better than the results of the ICSI baseline configuration for the MDM recordings for segment durations of 500 seconds and less.

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	12.13 %	-
	model1	12.85 %	+6.0 %
500	FastMatch (baseline)	15.21 %	-
	model1	12.83 %	-15.6 %
300	FastMatch (baseline)	25.02 %	-
	model1	13.68 %	-45.3 %
100	FastMatch (baseline)	44.00 %	-
	model1	17.91 %	-59.3 %

Table 5.5: Comparison of the performance of the linear regression model based on the seconds per Gaussian to the ICSI baseline system (FastMatch) for the 8 meetings of the RT-06 evaluation set (SDM recordings) and different split durations. Only the Speaker Error is presented. The speech/non-speech error is basically the same for both approaches and therefore not listed in the table.

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	9.89 %	-
	model1	11.23 %	+13.6 %
500	FastMatch (baseline)	14.92 %	-
	model1	12.8 %	-14.2 %
300	FastMatch (baseline)	18.74 %	-
	model1	12.54 %	-33.1 %
100	FastMatch (baseline)	34.59 %	-
	model1	18.77 %	-45.7 %

Table 5.6: Comparison of the performance of the linear regression model based on the seconds per Gaussian to the ICSI baseline system (FastMatch) for the 8 meetings of the RT-07 evaluation set (MDM recordings) and different split durations. Only the Speaker Error is presented. The speech/non-speech error is basically the same for both approaches and therefore not listed in the table.

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	15.53 %	-
	model1	13.26 %	-14.6 %
500	FastMatch (baseline)	17.88 %	-
	model1	14.51 %	-18.9 %
300	FastMatch (baseline)	22.41 %	-
	model1	14.07 %	-37.2 %
100	FastMatch (baseline)	37.76 %	-
	model1	17.55 %	-53.5 %

Table 5.7: Comparison of the performance of the linear regression model based on the seconds per Gaussian to the ICSI baseline system (FastMatch) for the 8 meetings of the RT-07 evaluation set (SDM recordings) and different split durations. Only the Speaker Error is presented. The speech/non-speech error is basically the same for both approaches and therefore not listed in the table.

were used for the RT-06 and RT-07 evaluation ($K = 16$ and $M = 5$). Analyzing the results in the tables, it can be seen, that the only configuration this engine performs significantly better, is the MDM complete meeting length configuration of the RT-07 evaluation. By having a closer look at the Speaker Errors of the individual meeting recordings (appendix A), this behavior may be explained mainly by the bad performance of the linear regression model on the recording *EDI_20061113-1500*. If this recording is excluded, the performance of the ICSI baseline system and the linear regression model are the same up to a 0.1% absolute difference (8.0143% for the ICSI baseline system and 8.1143% for the linear regression model). The reason of the bad behavior of the linear regression model on this particular meeting however remains unclear. For the complete meetings of the RT-06 evaluation data recorded with a Single Distant Microphone (SDM), the linear regression model performs slightly worse because it does not perform well on the meeting *EDI_20050218-0900*. It is also interesting to see on the NIST RT-07 evaluation set, that the linear regression model performance of the SDM recordings is better than the performance of the untuned engine of the MDM recordings for segment durations of 500 seconds and less. It can be concluded that the linear regression model is a good method to estimate the important paramter *seconds per Gaussian*, especially for short meetings. Furthermore, because of the very good behavior of the model on all the tested datasets, it seems to be generalizable. In the next section some concluding remarks on this linear regression model are given.

5.4.3 Concluding remarks on the linear regression model

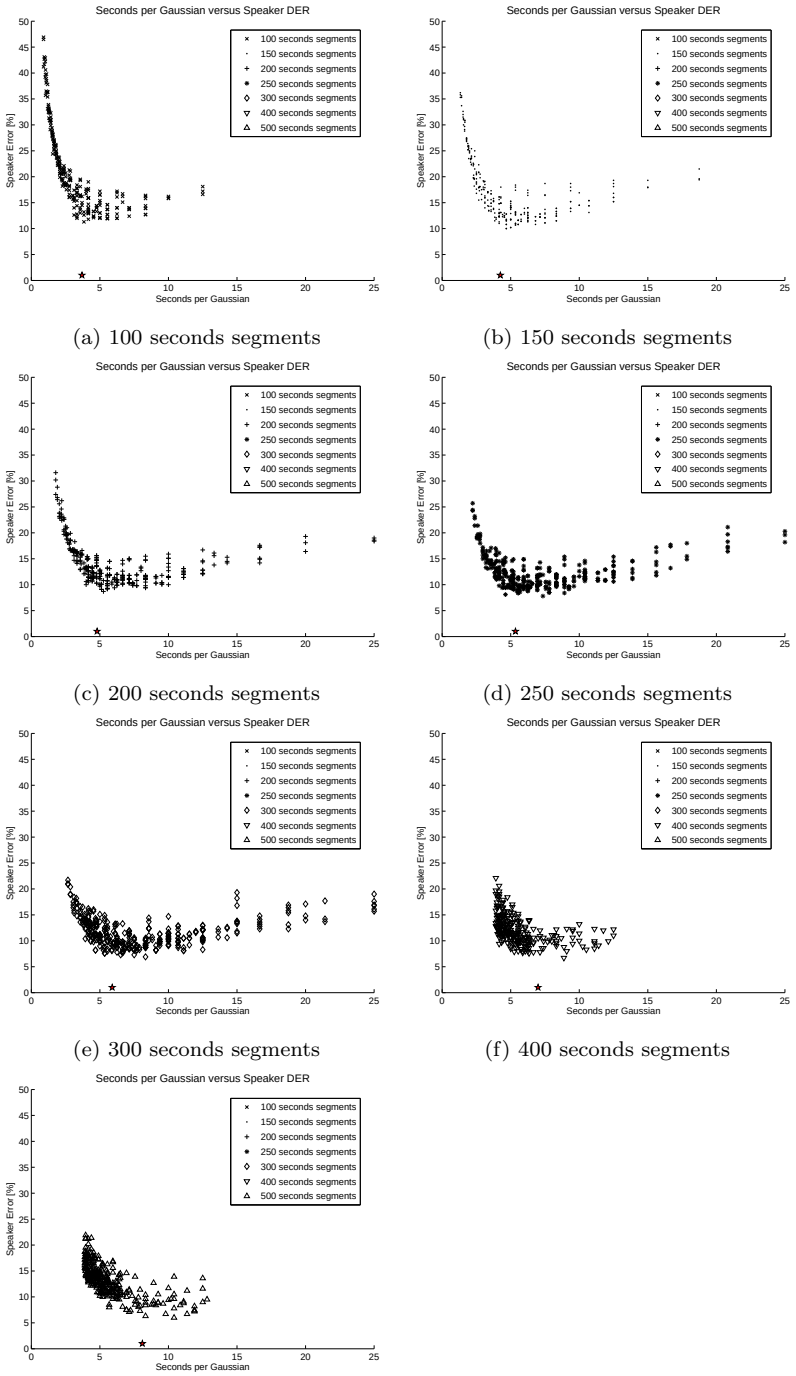
The linear regression model, based on the seconds per Gaussian is actually performing very well. It is interesting to see that the automated system outperforms the manually tuned ICSI baseline system. The very good performance can be explained by the fact that 4 Gaussians per initial cluster is a good overall choice (see Figure 5.4) and the linear regression that is based on the seconds per Gaussians seems to be a good trade-off between the total number of Gaussians used to model the data of the recording and the seconds of speech available to train one single Gaussian. A certain amount of Gaussians needs to be used to model the data in order to be able to distinguish between different speakers. If the GMMs with diagonal covariance to model the data consist of only one Gaussian for instance, there are only the mean and the variance vectors to decide if the clusters were produced by the same speaker or not. On the other hand, if there is not enough data to train the GMMs (made up of a certain number of Gaussians), the models will not fit the data and a comparison may be very difficult as well. By having a closer look at Figure 5.3 and analyzing the exhaustive search experiment results per segment duration (see Figure 5.10), it looks like there exists a global minimum for every segment duration. As the segment durations become longer, the minimum is moving to the right (more seconds per Gaussian). If that minimum is compared to the value of the linear regression for the seconds per Gaussian that is displayed as the star just above the x-axis, it is interesting to see that there is a very good overall fit between these values and they seem to move together along the x-axis. The global minimum moves to the right and so does the linear regression value which is actually determined by the following linear regression: $0.011x + 2.6$ where x is the speech duration in seconds (see Figure 5.5 (a)).

5.5 Visualizing the agglomerative clustering

In this section a graphical tool to visualize the agglomerative clustering process is presented and some conclusions of the visualization of the results are discussed.

5.5.1 The graphical tool

In oder to have a better understanding of the agglomerative clustering, a graphical tool to visualize the process was built. The tool that works together with the ICSI Speaker Diarization engine was designed in Matlab. In Figure 5.11 and 5.12, screenshots of the tool are shown. More data-specific explanations are given in Section 6.6, where the displayed data is discussed in more detail. The tool consists of three different plots:



(a) 100 seconds segments

(b) 150 seconds segments

(c) 200 seconds segments

(d) 250 seconds segments

(e) 300 seconds segments

(f) 400 seconds segments

(g) 500 seconds segments

Figure 5.10: Speaker error versus the seconds per Gaussian. Different segment durations are shown in every plot. The (global) minimum is moving to the right and so does the linear regression value (star above the x-axis, see Section 5.4.3).

1. **final results** Once the input files are loaded (ground truth, speech/non-speech output and system output) the tool shows the speech/non-speech segments (spnsp), the ground truth (speaker$n$-gt), the final output (speaker$n$-system) and temporary output (speaker$n$-progress, see point 2) of the engine for different speakers n. The x-axis represents the timeline and the different colors correspond to different speakers. In the ground truth it may be possible that more than one speaker is active at one point in time, this is a so-called overlap region. With the checkboxes, that are labeled with the ID of the speakers (given in the ground truth), on the right of the plot, a track n can be made visible or invisible by checking respectively un-checking the corresponding box. The two series of lists next to the checkboxes allows the user to manually choose the system output clusters (final or progress) that is aligned to a certain ground truth speaker. By clicking on the button play, it is possible to listen to a certain segment of the meeting (ground truth or system output).

2. **progress results** The tool also shows the clustering at different iteration steps. The x-axis represents the timeline in seconds and the y-axis the different iteration steps. The initial segmentation can be seen at the bottom and the final segmentation at the top of the plot. A different color is used per cluster and the colors do not at all correspond to the ones used in the *final results*-plot. At every iteration step, one cluster (color) disappears because it is merged with another cluster. On the right, a list is shown where the user can select a certain iteration step that is then plotted as temporary result in the *final results*-plot. In Figure 5.12 for example, the iteration step 16 (*015.rttm*) is chosen (the reason for that comparison is given in Section 6.6).

3. **eval info** Different error rates and the speaker mapping (as compared to the ground truth by the evaluation script) is given. By clicking on the button *apply mapping(s)*, the selected speaker mappings are applied to the system output in the *final results*-plot.

5.5.2 Visualization results

The graphical tool uncovered an unexpected initialization process: Given the *number of initial clusters* (K), the FastMatch-baseline implementation splits the data into $2K$ clusters and labels them with $1 \dots K\ 1 \dots K$. This special initialization apparently improved the performance during the tuning of the engine for an evaluation and finally explains why it was helpful to duplicate a short segment (the first experiments were done by using 100-second segments and duplicating them). Because of the duplication and the $1 \dots K\ 1 \dots K$ labeling of the clusters, the training is done twice on the exact same data identically segmented. As visualized in

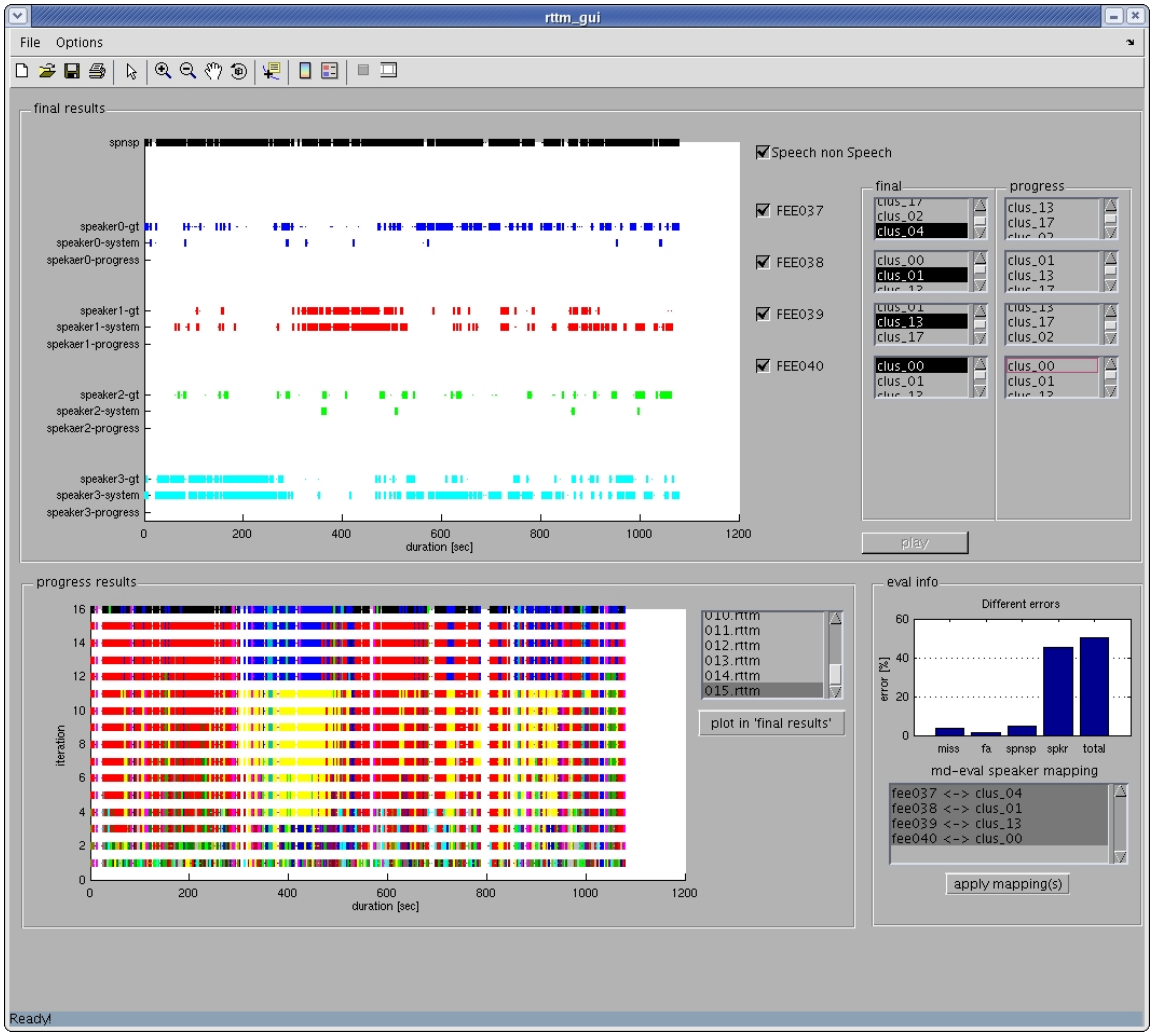


Figure 5.11: The graphical tool. (Tool related explanations: Section 5.5 and data related explanations (non-uniform initialization): Section 6.6.

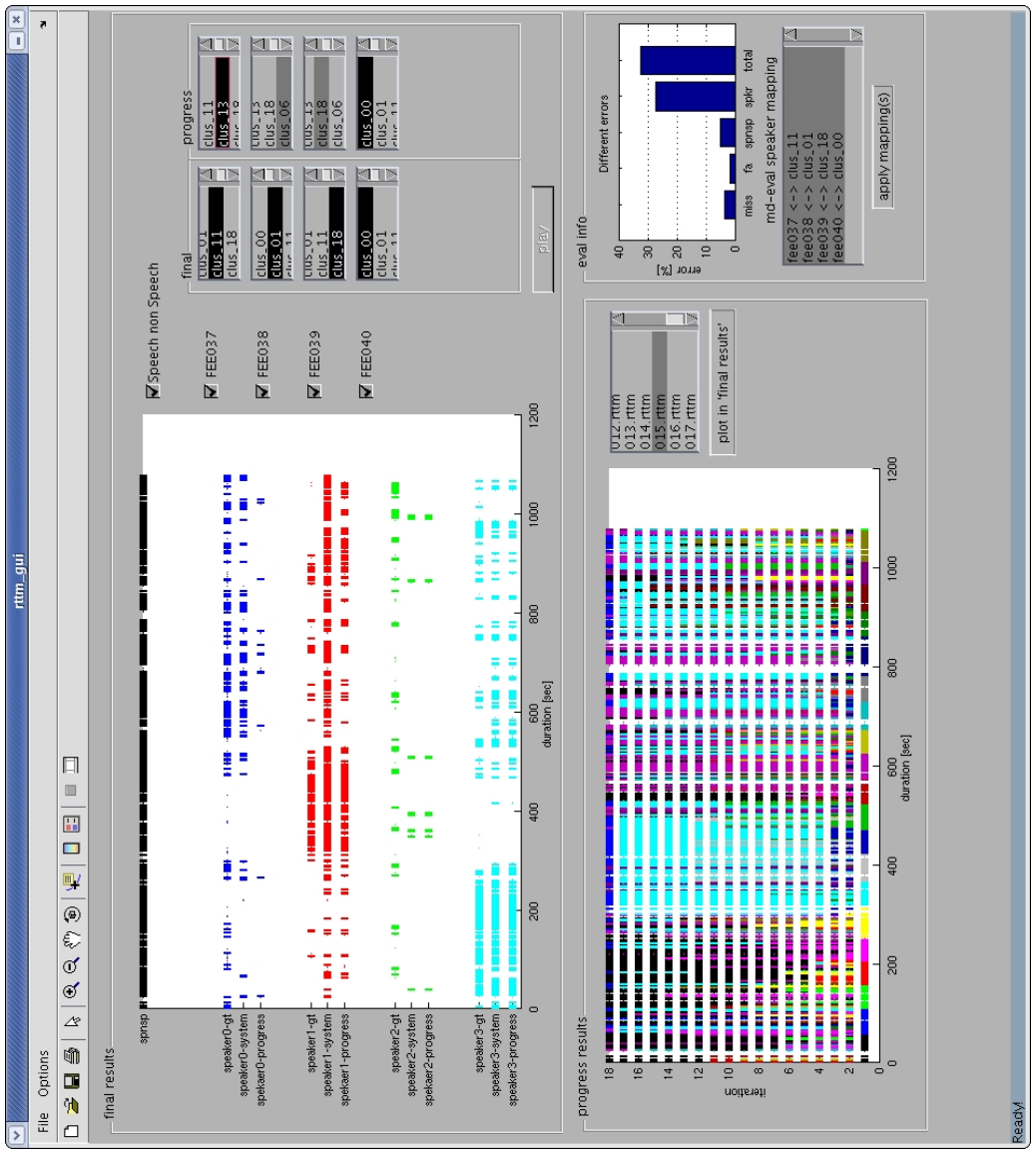


Figure 5.12: The graphical tool. (Tool related explanations: Section 5.5 and data related explanations (uniform initialization): Section 6.6.

Figure 5.13, a segment duplication in combination with this special initialization procedure is basically the same as doubling the number of GMM iteration of the training procedure and performing a differenet initialization (continuously uniform with labels $1 \dots K$). But as shown in Figure 5.14, only increasing the number of GMM iterations does not lower the Speaker Error as it is the case when duplicating segments. It can be seen that the Speaker Error fluctuates if the segment is duplicated several times and attains lower values always when the segment is an even number of times present. In this case, the split boundaries correspond to the initial segment boundaries. The fact that the duplication of the segment lowers the error more than increasing the number of GMM iterations shows that the initialization may have an effect on the final result. In the next chapter, a method to perform a non-uniform initialization will be presented.

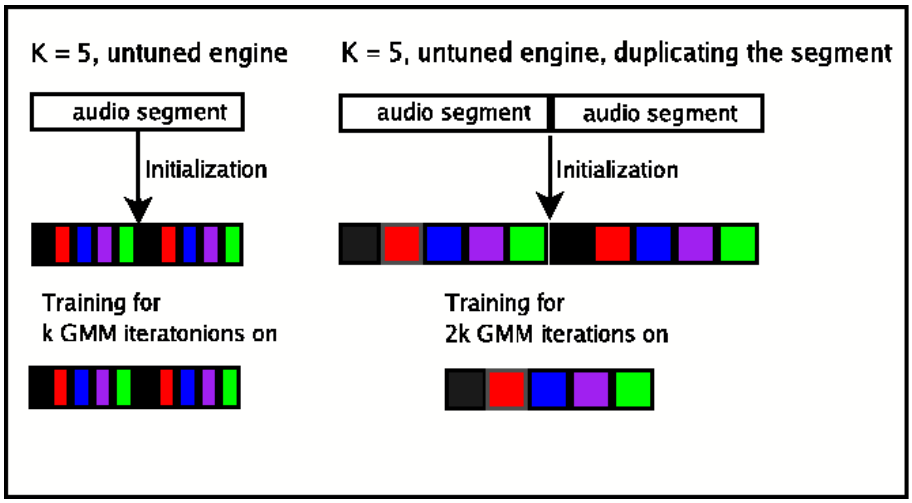
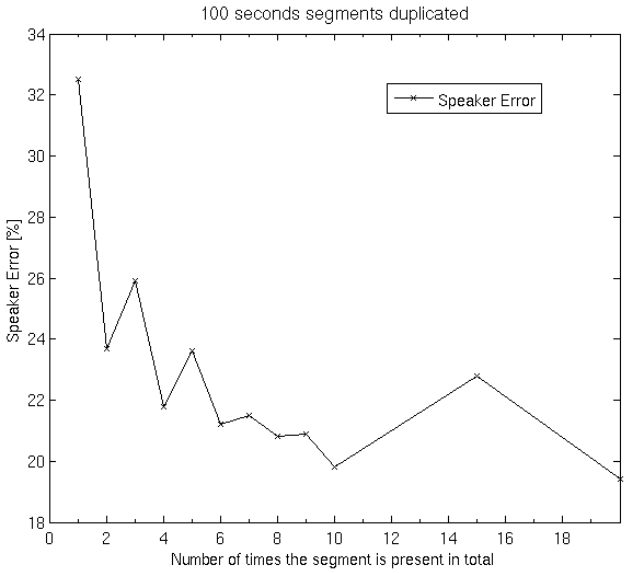
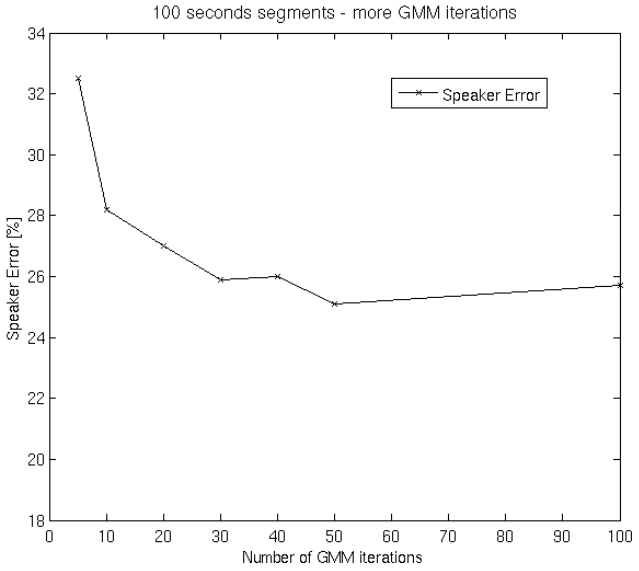


Figure 5.13: Visualization of the initialization process: During the tuning for an evaluation, the initialization process was changed to initially split the data into $2K$ clusters and labeling them with $1 \dots K$ $1 \dots K$. This effect was *hidden* in the code and discovered by the graphical tool and explains why the duplication of segments for some preliminary experiments improved the performance that much. Duplicating a segment and using this special initialization is basically the same as doubling the number of GMM iterations for the training and doing a continuously uniform initialization into K clusters.



(a) The y-axis represents the average Speaker Error in percent of the RT-06 development set split into 100-second segments and the x-axis shows the number of times a segment is present in total (see Section 5.5.2). The fluctuation can be explained by the fact that for even numbers, the split boundaries corespond to the initial segment boundaries (Section 5.5.2 and Figure 5.13)



(b) The y-axis represents the average Speaker Error in percent of the RT-06 development set split into 100-second segments and the x-axis shows how many GMM iterations are performed during the training phase.

Figure 5.14: Both plot show the average Speaker Error from the same data (RT-06 development set split into 100-second segments). Duplicating the segment ($x = 2$ in (a)) has a lower error ($y < 24\%$) than increasing the number of GMM iterations (in (b) $y \geq 24.5\%$ even for a large amount of iterations). Thus, the initialization has an effect on the final result.

Chapter 6

Better initialization methods

In the last chapter, an exhaustive search was performed where the whole audio track was split into segments of x seconds where $x \in \{100, 150, 200, 250, 300\}$ and a linear regression model was built based on the high correlation between different parameters among the best performing results of every segment duration. Analyzing the 10 best-performing configurations per duration instead of only taking the best one into consideration, leads to a very high correlation between the speech duration and the total number of Gaussians (the number of initial clusters K multiplied with the number of Gaussians per cluster M, for more details, see Section 6.3). This parameter is the multiplication of two tunable parameters, thus the high correlation cannot be exploited with the help of a trivial linear regression. Furthermore, just fixing one parameter to a static value as it was done in the last chapter is unsatisfying. Having in mind that the initialization of the clustering engine may have an influence on the final result and the secondary goal is to get rid of parameters, this chapter presents and tests a method to not only get rid of both parameters (the number of initial cluster and the number of Gaussians per initial cluster) by estimating them in an appropriate way but also to do a better performing non-uniform initialization.

6.1 Estimation of the number of initial clusters

One possibility to estimate the number of initial clusters is to use prosodic features in combination with a model selection procedure. Prosodic features have been successfully combined with MFCCs to do Speaker Diarization at ICSI before and many different prosodic features and other long-term features have been studied and ranked according to their speaker discriminate ability. Based on the ranking method proposed in [Friedland et al., 2009b], 12 top-ranked prosodic features (listed in Table 6.1) are extracted on all the speech regions in the recording. For the extraction of the prosodic features, a library named *praatlib* that is using Praat [Praat, 2009] functions, was developed at ICSI. For more detailed information about the different features and extraction methods, the reader is referred to the documentation of

feature category	feature id	short description
pitch	f0_median	median of the pitch on a specified segment
pitch	f0_min	minimum of the pitch on a specified segment
pitch	f0_mean_curve	mean of the pitch tier (a time-stamped pitch contour) on a specified segment
formants	f4_stddev	standard deviation of the 4th formant on a specified segment
formants	f4_min	minimum of the 4th formant on a specified segment
formants	f4_mean	mean of the 4th formant on a specified segment
formants	f5_stddev	standard deviation of the 5th formant on a specified segment
formants	f5_min	minimum of the 5th formant on a specified segment
formants	f5_mean	mean of the 4th formant on a specified segment
harmonics	harm_mean	mean of the harmonics-to-noise ratio on a specified segment
formant	form_disp_mean	mean of the formant dispersion on a specified segment
pitch	pp_period_mean	mean of the pointprocess of the periodicity contour on a specified segment

Table 6.1: These 12 prosodic features have a good speaker discriminate ability according to the ranking method proposed in [Friedland et al., 2009b]. The features are extracted with the help of praatlib, a library that is using Praat [Praat, 2009], on all the speech regions of the recordings and afterwards used to estimate the number of initial clusters to perform the agglomerative clustering. For more information about the features refer to the documentation of Praat.

Praat. The 12-dimensional feature vectors are clustered with the help of one GMM with diagonal covariance. The model selection criterion is kept as simple as possible and the number of Gaussians the GMM consists of is determined by comparing the negative log-likelihoods of GMMS with different number of Gaussians. Having in mind that the resulting clustering serves as input for an agglomerative clustering algorithm and more elaborate model selection criteria punish more complex models, using the negative log-likelihood seems to be an adequate choice. It is desired that the model selection tends to over-estimate the number of initial clusters as the agglomerative clustering algorithm will merge redundant clusters whereas it is not able to split clusters. For the clustering and model selection, the openly available clustering source code of Weka [Weka, 2008] is modified. Basically, a 10-fold cross-validation is used to calculate the log-likelihood of a GMM with a certain number of Gaussians. It is decided how many Gaussians to use and then the Expectation Maximation algorithm (EM, [Bishop, 1995], p. 65) is used to train the GMM with the specified number of Gaussians on all the feature vectors extracted on the speech segments of the complete recording. Finally, the clustering assigns every feature vector to a certain initial segment and results in a non-uniform initialization where the number of initial clusters is automatically determined. The main idea behind this method is that the engine benefits from putting similar regions (similar in terms of features that have the ability of discriminating speakers) together for the initialization and that the model selection criterion estimates an appropriate choice for the number of initial clusters.

6.2 Prosodic feature extraction

In order to get accurate feature vectors, some technical issues concerning the extraction process are discussed in this section. To perform the prosodic feature extraction on the speech segments only, there are different possibilities: the features can be extracted on the segments found by the speech/non-speech detector, what may result in very few feature vectors for the clustering because the segments are relatively large compared to typical window size choices. In [Friedland et al., 2009b] for example, a 500-ms hamming window with overlap is used to extract the features. In general, the long-term prosodic feature calculations are more accurate if the window is longer, but for the estimation of the number of initial clusters K and the clustering itself, a certain amount of feature vectors is needed to result in a good estimation of K and a reasonable non-uniform initialization. For this work, the hamming windowing function is used and two different minimal window lengths are considered to test the proposed method: 500 ms and 1000 ms. A minimal window size of 500 ms means that every segment (output of the speech/non-speech detector) of less than 1000 ms is untouched and the larger ones are split into segments of at

least 500 ms (effectvive window length $w \in [500, 1000[$ for a minimal window size of 500 ms).
With prosodic and long-term features it may happen that for a certain segment some features such as pitch are undefined. In that case, the value is set to the mean value of that feature, calculated over all the feature vectors of the corresponding audio recording.

6.3 Choosing the number of Gaussians per initial cluster

There are different ways of estimating the number of Gaussians per initial cluster automatically. In the last chapter, the number was statically fixed to four what performed relatively well but may only work for these particular sets and is unsatisfactory. Once the number of initial clusters is estimated, the number of Gaussians per initial clusters can be determined by using the linear regression model based on the seconds per Gaussian that performed well (see Section 5.4.3) or by using a linear regression that is based on the relation between speech duration and the total number of Gaussians. This relation has a higher correlation value (0.8226) than the seconds per Gaussian relation (0.6756). The correlation values are calculated as mentioned in Section 5.3. The linear regression, calculated in Matlab (see Figure 6.1), is $totgauss = 0.1x + 19$, where x is the speech duration.

6.4 Testing the proposed method

In Section 6.1, a method to estimate the number of initial clusters and performing a non-uniform initialization is proposed and in Section 6.3, methods to estimate the number of Gaussians per initial clusters are proposed. These methods are tested on different data-sets and all the results are presented in this section.

6.4.1 Development set RT-06

In Table 6.2, the results for processing complete meetings of the RT-06 development set are shown. The ICSI baseline system is the FastMatch engine (ICSI engine that uses fast-match component, presented in [Huang et al., 2007]) with $K = 16$ and $M = 5$. The average error of the 12 meetings from the RT-06 development set is shown.
Two different minimal window lengths (500 ms and 1000 ms) and two different linear regression models to determine the number of Gaussians (seconds per Gaussian and total Gaussians) are compared. The average number of initial cluster that are found

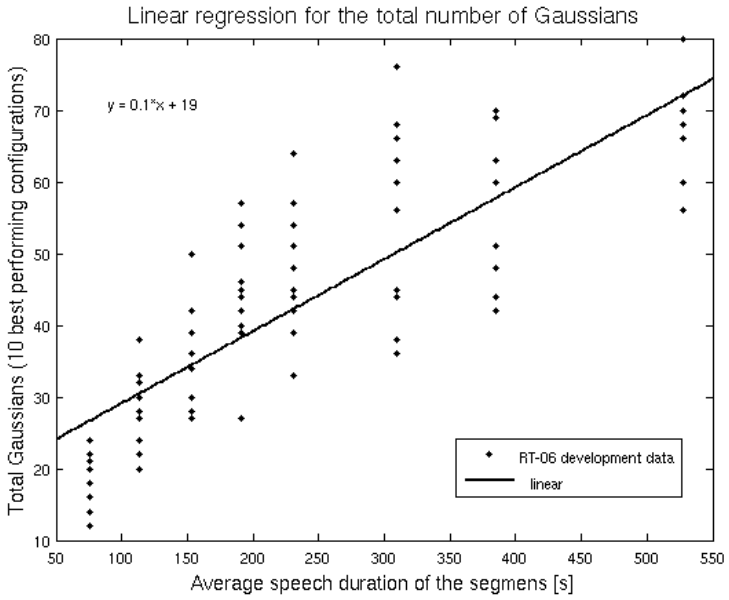


Figure 6.1: The average speech duration of the segments (x-axis) versus the total number of Gaussians for the 10 best performing configurations per segment duration. The relation shows a relatively high correlation and the linear regression (calculated with Matlab) is also shown in the plot.

Configuration	Window size	Average K	Gaussians estimation	Speaker Error	Relative change
baseline	-	16	5	7.40 %	-
spg 500	500 ms	20.33	sec/gauss	10.46 %	+41.3 %
spg 1000	1000 ms	15.00	sec/gauss	6.97 %	-5.9 %
tg 500	500 ms	20.33	tot gauss	8.56 %	+15.7 %
tg 1000	1000 ms	15.00	tot gauss	7.17 %	-3.2 %

Table 6.2: Comparison of different configurations for the development set of the RT-06 evaluation. The configuration description consists of the linear regression that was used, where *spg* stands for seconds per Gaussian and *tg* stands for total Gaussians and the number is the minimal window size in milliseconds used for the prosodic feature extraction.

by the clustering algorithm are also displayed. This estimation only depends on the minimal window length (different prosodic features result from a different minimal window length). Both configurations with the 1000 ms window perform slightly better than the actual ICSI baseline system, but it needs to be stated that the parameters were manually tuned to $K = 16$ and $M = 5$ using this development set, thus it is quite unexpected that a configuration without manually tuned parameters performs equally well. Interestingly, the average estimated number of clusters is 15 for the 1000 ms minimal window length, which is very close to the manually tuned value (16). However this shows that the window size of the prosodic feature extraction influences the final result and 1000 ms seems to be the better choice. This hypothesis will further be tested on other data-sets.
In Figure 6.2, the results for different segment durations are displayed (100 s, 300 s, 500 s and complete meetings). The curves that are labeled with prosodic features initialization (prosodic init) have also specified the linear regression model that was used (seconds per Gaussian or total number of Gaussians) and the window size that was used (500 ms or 1000 ms). It can be seen that all the version without manually tuned parameters outperform the manually tuned FastMatch engine on segment durations of 500 seconds and less. The configurations with the smaller window size (500 ms) have a lower average Speaker Error for shorter segment durations what can be explained by the fact that the shorter minimal window size results in more feature vectors to perform a good clustering and therefore do a better estimation of K. In general, a longer minimal window sizes results in more accurate values for long-term and prosodic features, but it seems that there are too few vectors to do a reasonable clustering with the 1000 ms minimal window length on shorter segments. It can also be seen that the linear regression model choice has only very small influence on the average Speaker Error on that data-set.

6.4.2 Evaluation set RT-06 MDM condition

In Table 6.3, the results for processing complete meetings of the RT-06 evaluation set under the MDM condition are shown. Again, the ICSI baseline system is the FastMatch engine with $K = 16$ and $M = 5$ and the average error of the 9 meetings is shown. It can be seen that all the configurations without manually tuned parameters are performing better than the baseline system. Furthermore, the best performing configuration is again the one that is based on 1000 ms minimal window length for the prosodic feature extraction and the linear regression based on the seconds per Gaussian. The 1000 ms minimal window size results in 18.33 initial clusters on average what is more than 16 (used in the manually tuned system). This time, the configuration with 1000 ms minimal window size and seconds per Gaussian regression performs not only significantly better than all other configurations,

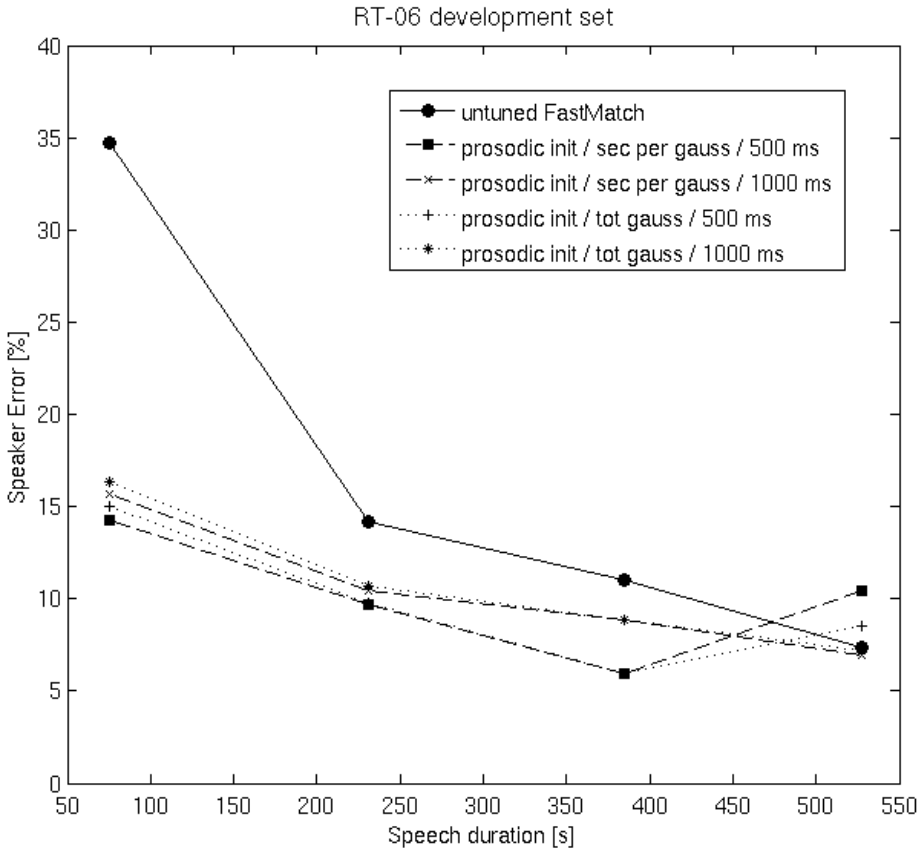


Figure 6.2: Comparison of four different configurations to the ICSI baseline engine for the RT-06 development set. All four configurations use prosodic initialization and any combination of 500 ms or 1000 ms minimal window size and seconds per Gaussian oder total Gaussians linear regression model (see 6.4.1). Every data point represents the average Speaker Error in percent (y-axis) of the 12 meetings from the RT-06 development set split into segments of a certain duration. On the x-axis the average speech duration is shown. The configurations with the 500 ms window perform better on shorter segments whereas the ones with 1000 ms window perform better on the longer segment durations.

but outperforms the ICSI baseline system with a relative improvement of 30%. In

Configuration	Window size	Average K	Gaussians estimation	Speaker Error	Relative change
baseline	-	16	5	16.68 %	-
spg 500	500 ms	21.77	sec/gauss	14.21 %	-14.8 %
spg 1000	1000 ms	18.33	sec/gauss	11.64 %	-30.2 %
tg 500	500 ms	21.77	tot gauss	14.51 %	-13.0 %
tg 1000	1000 ms	18.33	tot gauss	14.22 %	-14.7 %

Table 6.3: Test of different configurations for the RT-06 evaluation set (MDM condition). The configuration description consists of the linear regression that was used, where *spg* stands for seconds per Gaussian and *tg* stands for total Gaussians and the number is the minimal window size in milliseconds used for the prosodic feature extraction.

Figure 6.3, the results for different segment durations are displayed. In that graph, the remarkable better performance of the configuration with 1000 ms window and the seconds per Gaussian linear regression model for complete meetings is visible. It can again be recognized that the configurations with 500 ms widows for the feature extraction performs better on shorter meetings. The 500 ms window size may result in less accurate feature values, but produces much more vectors to do the clustering and the estimation of K. It can be seen that the curves of the 500 ms windows and the 1000 ms windows intersect, but the linear model choice to estimate the number of Gaussians has not that much influence for shorter segment durations.

6.4.3 Evaluation set RT-07 MDM condition

So far, the prosodic initialization method was tested on the development and evaluation set of the NIST RT-06 evaluation. To conclude which of the configurations performs best, some more experiments on the NIST-RT07 evaluation set (MDM condition) are performed. In Table 6.4, the results for processing complete meetings are shown. The baseline system is the FastMatch engine with $K = 16$ and $M = 5$, the average error of the 8 meetings from the RT-07 evaluation set is shown. Three of the four configurations without manually tuned parameters perform better than the ICSI baseline system. This time the configuration with the 1000 ms minimal window size and the total Gaussians regression performs best. However, also the configuration with 1000 ms minimal window size and the seconds per Gaussian regression performed very well and significantly better than the ICSI baseline system. For this minimal window size, the clustering process estimated 18.62 initial

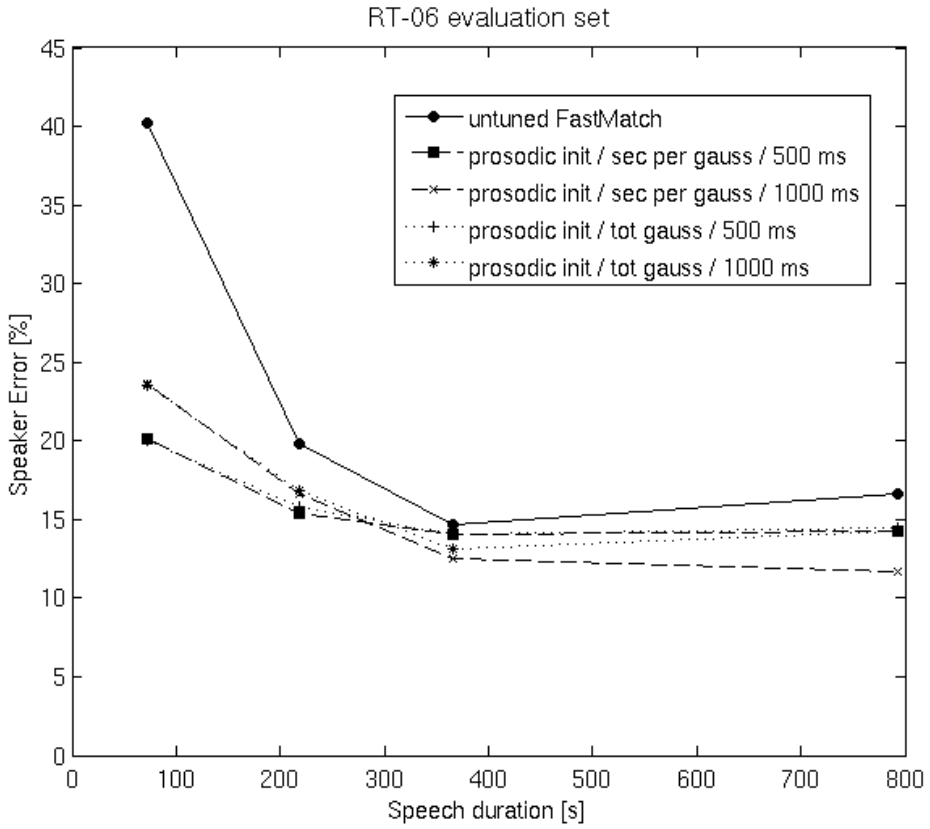


Figure 6.3: Comparison of four different configurations to the ICSI baseline engine for the RT-06 evaluation set (MDM condition). All four configurations use prosodic initialization and any combination of 500 ms or 1000 ms minimal window size and seconds per Gaussian or total Gaussians linear regression model (see 6.4.1). Every data point represents the average Speaker Error in percent (y-axis) of the 9 meetings from the RT-06 evaluation set split into segments of a certain duration. On the x-axis the average speech duration is shown. The configurations with the 500 ms window perform better on shorter segments whereas the ones with 1000 ms window perform better on the longer segment durations.

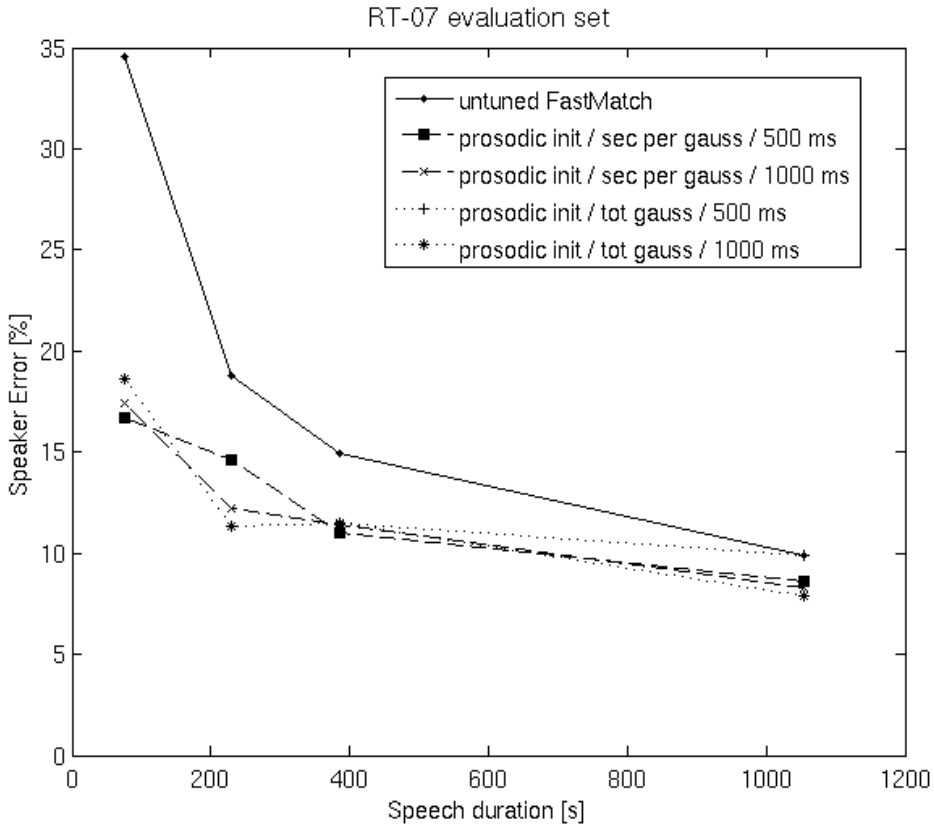


Figure 6.4: Comparison of four different configurations to the ICSI baseline engine for the RT-07 evaluation set (MDM condition). All four configurations use prosodic initialization and any combination of 500 ms or 1000 ms minimal window size and seconds per Gaussian oder total Gaussians linear regression model (see 6.4.1). Every data point represents the average Speaker Error in percent (y-axis) of the 8 meetings from the RT-07 evaluation set split into segments of a certain duration. On the x-axis the average speech duration is shown. There is not so much difference between the different configurations, but all perform better than the manually tuned ICSI baseline system.

Configuration	Window size	Average K	Gaussians estimation	Speaker Error	Relative change
baseline	-	16	5	9.89 %	-
spg 500	500 ms	21.87	sec/gauss	8.63 %	-12.8 %
spg 1000	1000 ms	18.62	sec/gauss	8.25 %	-16.6 %
tg 500	500 ms	21.87	tot gauss	9.91 %	+0.2 %
tg 1000	1000 ms	18.62	tot gauss	7.90 %	-20.1 %

Table 6.4: Test of different configurations for the RT-07 evaluation set (MDM condition). The configuration description consists of the linear regression that was used, where *spg* stands for seconds per Gaussian and *tg* stands for total Gaussians and the number is the minimal window size in milliseconds used for the prosodic feature extraction.

clusters on average.
In Figure 6.4, the results for shorter segments are displayed. This graph looks different from the one for the RT-06 evaluation set (Figure 6.3) in terms of minimal window size effects on the average Speaker Error. All prosodic initialization methods perform very similar and there is not much difference in the average Speaker Errors. Nevertheless, the configurations without manually tuned parameters perform better than the ICSI baseline system. In Table 6.5, the average estimated number of initial clusters for the three data-sets (dev06, eval06 and eval07) is compared to the average number of speakers in the ground truth and the average speech duration in the meetings in seconds. On one hand, the estimated number of initial clusters does not correlate well with the number of speakers in the ground truth. It can be seen that eval06 has most speakers on average but the algorithm did not estimate most initial clusters on average. On the other hand, eval07 has the longest speech duration and most estimated clusters on average, but the difference of the estimated number of initial clusters on average between eval06 and eval07 is very small. However, the clustering algorithm does not estimate the number of speakers in the meeting, but groups similar regions in terms of prosodic features together. In general, more speakers in the ground truth and more speech in the meeting tend to increase the estimated number of initial clusters.
Based on the results on the RT-06 development set and the evaluation sets of RT-06 and RT-07 (MDM condition), it is decided that the configuration with a 1000 ms minimal window size and the seconds per Gaussian linear regression model is further tested and analyzed because it has the best overall performance and the other configurations are discarded. To ensure that the novel initialization procedure performs better than the manually tuned actual ICSI baseline, some more experiments

are performed on the evaluation sets of RT-06 and RT-07 under the SDM condition and 12 chosen AMI meetings. In the next section, tables with the exact Speaker Error rates are given to compare the ICSI baseline system to the novel initialization method.

Data-set	Average estimated number of initial clusters	Average number of speakers	Average speech duration [s]
dev06 MDM	15.00	4.42	527.42
eval06 MDM	18.33	5.11	793.40
eval07 MDM	18.62	4.38	1053.96

Table 6.5: Comparison of the average estimated number of initial clusters (with the clustering method described in Section 6.1), the average number of speakers (in the ground truth) and the average speech duration of the RT-06 development set (dev06) RT-06 MDM evaluation set (eval06) and RT-07 MDM evaluation set (eval07). There is no evident relation between any of these parameters.

6.5 Results

In this section, the prosodic initialization method using a 1000 ms minimal window size to perform the prosodic feature extraction and the seconds per Gaussian linear regression model to estimate the number of Gaussians per initial cluster is compared to the ICSI baseline system on different data sets and for different segment durations (see Tables 6.6 to 6.11). Only the Speaker Error is listed in the tables, because the speech/non-speech error is basically the same for the ICSI baseline system and the novel initialization method. In appendix A, the Speaker Errors of the individual meetings are listed. The novel method lowers the Speaker Error especially for shorter meetings by a large amount (up to 60 percent relative improvement) but even in the complete meetings case, the novel initialization method performs better than the ICSI baseline system (up to 30 percent relative improvement) on all sets except the evaluation set of the RT-06 when the Single Distant Microphone (SDM) recordings are used. Averaging the Speaker Errors of all the tested meetings (development set RT-06, evaluation sets RT-06 and RT-07 SDM and MDM and 12 chosen AMI meetings) results in an error of 12.49% for the ICSI baseline system and 10.30% for the novel initialization method which is a relative improvement of 17.3 percent. Therefore, it can be concluded that the new initialization method

performs significantly better than the ICSI baseline system which performs only on the NIST RT-06 evaluation set and SDM condition better than the novel method.

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	7.40 %	-
	prosodic initialization	6.97 %	-5.9 %
500	FastMatch (baseline)	11.02 %	-
	prosodic initialization	8.85 %	-19.7 %
300	FastMatch (baseline)	14.18 %	-
	prosodic initialization	10.40 %	-26.7 %
100	FastMatch (baseline)	34.74 %	-
	prosodic initialization	15.66 %	-54.9 %

Table 6.6: Comparison of the prosodic initialization method to the ICSI baseline system which was manually tuned. On the development set of the RT-06, the novel initialization methods outperforms the baseline for every segment duration. The relative improvement is more than 50 percent for 100-second segments.

6.6 Influence of the initialization

At the end of Chapter 5, it was claimed that the initialization may have an influence on the final result. In this chapter, it was shown so far, that the proposed method improves the result by combining completely unsupervised parameter estimation with a non-uniform initialization, but what happens if the estimated parameter values for the number of initial cluster K and the Gaussians per initial clusters M are used to do a uniform initialization? The results are listed in Table 6.12 and it can be seen that the initialization affects the performance significantly. In one only case, for the evaluation set of the year 2006 under the Single Distant Microphone (SDM) condition, the performance is better when a uniform initialization is used. This is mainly due to the bad performance of the non-uniform initialization on the meeting *EDI_20050216-1051*. In Figure 5.11, the clustering process for the non-uniform initialization is shown and in Figure 5.12, the process for the uniform initialization. It can be seen that the non-uniform initialization clustering stops with six clusters, whereas the uniform initialization process correctly ends up with four clusters (four speakers in the ground truth). In Figure 5.12, the temporary result (that is plotted in the *final results*-plot) corresponds to the segmentation at the iteration step with 6 clusters and it can be seen that the segmentation at that iteration step is similar to the final segmentation of the non-uniform initialization. For that particular meeting,

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	16.68 %	-
	prosodic initialization	11.64 %	-30.2 %
500	FastMatch (baseline)	14.70 %	-
	prosodic initialization	12.47 %	-15.2 %
300	FastMatch (baseline)	19.82 %	-
	prosodic initialization	16.63 %	-16.1 %
100	FastMatch (baseline)	40.23 %	-
	prosodic initialization	23.60 %	-41.3 %

Table 6.7: Comparison of the prosodic initialization method to the ICSI baseline system. On the evaluation set of the RT-06 and the MDM condition. The novel initialization methods outperforms the ICSI baseline for every segment duration. The relative improvement of 30 percent for complete meetings is worth mentioning.

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	12.13 %	-
	prosodic initialization	14.88 %	+22.7 %
500	FastMatch (baseline)	15.21 %	-
	prosodic initialization	11.99 %	-21.1 %
300	FastMatch (baseline)	25.02 %	-
	prosodic initialization	15.29 %	-38.9 %
100	FastMatch (baseline)	44.00 %	-
	prosodic initialization	19.91 %	-54.8 %

Table 6.8: Comparison of the prosodic initialization method to the ICSI baseline system. On the evaluation set of the RT-06 and the SDM condition. The novel initialization method performs worse than the ICSI baseline system for complete meetings. That is the only condition, the novel initialization method performs worse and this case is studied in more detail in Section 6.6.

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	9.89 %	-
	prosodic initialization	8.25 %	-16.6 %
500	FastMatch (baseline)	14.92 %	-
	prosodic initialization	11.42 %	-23.4 %
300	FastMatch (baseline)	18.74 %	-
	prosodic initialization	12.24 %	-34.7 %
100	FastMatch (baseline)	34.59 %	-
	prosodic initialization	17.41 %	-49.7 %

Table 6.9: Comparison of the prosodic initialization method to the ICSI baseline system. On the evaluation set of the RT-07 and the MDM condition. The novel initialization method outperforms the ICSI baseline system with a continuously growing relative improvement towards shorter segment durations and reaches almost 50 percent relative improvement for 100-second segments.

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	15.53 %	-
	prosodic initialization	10.30 %	-33.7 %
500	FastMatch (baseline)	17.88 %	-
	prosodic initialization	13.46 %	-24.8 %
300	FastMatch (baseline)	22.41 %	-
	prosodic initialization	15.41 %	-31.3 %
100	FastMatch (baseline)	37.76 %	-
	prosodic initialization	16.26 %	-56.9 %

Table 6.10: Comparison of the prosodic initialization method to the ICSI baseline system. On the evaluation set of the RT-07 and the SDM condition, the novel initialization method outperforms the ICSI baseline system on all segment durations with a maximum relative improvement of almost 57 percent for 100-second segments.

Segment duration	Configuration	Speaker Error	Relative change
tot	FastMatch (baseline)	14.38 %	-
	prosodic initialization	10.95 %	-23.9 %
500	FastMatch (baseline)	12.98 %	-
	prosodic initialization	9.72 %	-25.1 %
300	FastMatch (baseline)	19.04 %	-
	prosodic initialization	12.35 %	-35.4 %
100	FastMatch (baseline)	36.22 %	-
	prosodic initialization	13.46 %	-62.4 %

Table 6.11: Comparison of the prosodic initialization method to the ICSI baseline system. On 12 chosen AMI meetings, the novel initialization method outperforms the ICSI baseline system on all segment durations significantly. For 100-second segments, the relative improvement is 62.4 percent on average.

the non-uniform initialization influences rather the decision of the stopping criterion than the segmentation what results in an enormous difference in the Speaker Error (45.3% for the non-uniform initialization and 27.4% for the uniform initialization). This bad performace also affects the average Speaker Error. For all the other data sets, the non-uniform initialization performs significantly better than the uniform one and the ICSI baseline engine. Averiging the Speaker Errors of all the tested meetings (development set RT-06, evaluation sets RT-06 and RT-07 SDM and MDM and 12 chosen AMI meetings) results in an error of 12.49% for the baseline system. Using the estimated parameter values for the number of initial clusters K and the number of Gaussians per initial cluster M, but performing a uniform initialization instead of the novel non-uniform one results in 11.46% Speaker Error on average. Having in mind the results with the non-uniform initialization (10.30% Sepaker Error on average) leads to the conclusion that a relative improvement of 10.1 percent is caused only by the changed initialization procedure.

Data set	Baseline	Uniform initialization	Relative change	Prosodic initialization	Relative change
dev06 MDM	7.40 %	7.19 %	-2.8 %	6.97 %	-5.8 %
eval06 MDM	16.68 %	16.23 %	-2.7 %	11.64 %	-30.2 %
eval06 SDM	12.13 %	10.66 %	-12.1 %	14.88 %	+22.7 %
eval07 MDM	9.89 %	10.90 %	+10.2 %	8.25 %	-16.6 %
eval07 SDM	15.53 %	12.89 %	-17.0 %	10.30 %	-33.7 %
AMI	14.38 %	12.11 %	-15.8 %	10.95 %	-23.9 %
ALL	12.49 %	11.46 %	-8.3 %	10.30 %	-17.3 %

Table 6.12: Influence of the initialization on the result for tests on the RT-06 development set (dev06) RT-06 evaluation set (eval06), RT-07 evaluation set (eval07) and 12 chosen AMI meetings (AMI). The Speaker Errors are listed for the ICSI baseline system, the system with prosodic initialization and a system that uses the same parameter values for the number of initial clusters K and the Gaussians per initial cluster M as the prosodic initialization system, but a uniform initialization.

Chapter 7

Limits of the Approach

The improvements, presented in the previous chapters, have shown that the proposed methods perform well. However, there are certain limitations to them, as outlined in this chapter.

7.1 Number of speakers

As explained in Chapter 1, the number of speakers is by task definition not known. Predicting the correct number of speakers has a significant impact on the overall accuracy [Fiscus and Ajot, 2007] but is difficult, especially in the case of agglomerative clustering where one merge more or less may change the Speaker Diarization Error Rate by several percent absolute (see Section 6.6 for an example). This is one of the reasons why the engine is that sensitive to small parameter changes and is a general problem of the agglomerative clustering approach.

7.2 BIC as decision criterion

The Bayesian Information Criterion (BIC) method certainly works very well. However the decision that this criterion makes may be suboptimal. The decision of ΔBIC (see Section 3.3) is not only influenced by the parameters as can be seen in Table 7.1 but also different initialization methods may affect the decision (see Section 6.6). In Table 7.1, the results of the following experiment is displayed: the ground truth serves as input for the clustering algorithm. In the ideal case, the clustering algorithm should do nothing and just give the ground truth as output, but the algorithm does merge clusters in some cases. Increasing the total number of Gaussians affects the result.

Meeting ID	Number of speakers in ground truth	Number of speakers found (80 Gaussians in total)	Number of speakers found (150 Gaussians in total)	Number of speakers found (200 Gaussians in total)
CMU_20050301-1415	4	2	3	3
ICSI_20000807-1000	6	4	4	4
ICSI_20010208-1430	6	5	5	4
LDC_20011116-1400	3	2	3	3
NIST_20030925-1517	4	3	3	3
VT_20050318-1430	5	3	5	5

Table 7.1: Behavior of the ΔBIC criterion (see Section 3.3) if the total number of Gaussians are changed and the ground truth is given as input to the clustering engine for chosen meetings of the development set of the NIST RT-06 evaluation. The engine merges clusters, but the ground truth contains the correct number of speakers. Further, it can also be seen that varying the total number of Gaussians affects the decision of the ΔBIC criterion.

7.3 Clustering with few samples

The novel initialization method proposed in Chapter 6 works well for the complete meetings case. If the meetings are split into smaller pieces however, the performance may be increased by changing the minimal window size of the prosodic feature extraction process. The estimation of the number of initial clusters is not very accurate if there is only a small number of feature vectors available to do the clustering. If the window size used during the prosodic feature extraction is smaller, more feature vectors are available and the results are better for shorter segment durations. In the end, a trade-off has to be made between having many vectors to perform clustering and having the more accurate long-term and prosodic feature values.

Chapter 8

Conclusion and future work

In Chapter 4, it was shown that the current ICSI Speaker Diarization engine performs suboptimal on shorter segment durations (Figure 4.4). Therefore the initial goal was to improve the performance on short segment durations and to possibly reduce the number of manually tunable parameters. In this work, methods were discussed to reduce the number of manually tunable parameters and get a more accurate Speaker Diarization result at the same time. This result is valid not only for short segment durations but also for complete meetings. In Section 5.4.3, the importance of the *seconds per Gaussian*-parameter for the agglomerative clustering approach to Speaker Diarization is underlined. This parameter describes how many seconds of speech are available to train one single Gaussian. If the value of the parameter is too small, there is not enough speech available to train the system and if the value is too high there may be not enough Gaussians per speaker model to be able to separate different speakers accurately. As shown in Chapter 5, the linear regression that was built results in a good trade-off between having enough speech per Gaussian and having enough Gaussians. The analysis of the behavior of this parameter may also be helpful for other Speaker Diarization systems using GMM/HMM-based approaches in combination with the Bayesian Information Criterion.
The linear regression based on the *seconds per Gaussian* is combined with a novel initialization method that is based on prosodic features. The resulting engine outperforms the current ICSI GMM/HMM-based approach using agglomerative clustering significantly. A clustering approach estimates the number of initial clusters (Section 6.1). This estimated number is not an estimation of the number of speakers in the meeting, but the number that maximizes the negative log-likelihood of the clustering of all the 12-dimensional prosodic feature vectors with a GMM with diagonal covariance. Thus, this approach groups speech regions that are similar in terms of prosodic features with the ability of discriminating speakers together and overestimates the number of speakers in the meeting. This result is used to perform a non-uniform initialization and it was shown in Section 6.4 that this novel approach performs better than the current approach at ICSI. Furthermore, in Sec-

tion 6.6 it was shown that the initialization procedure affects the final result what may be generalizable to other agglomerative clustering approaches. However, the improvement to expect may be limited, depending on the system. An important contribution of this work to the ICSI Speaker Diarization engine is the accurate estimation of sensitive parameters what is not generalizable to other systems.
In Table 8.1, the novel initialization method is compared to the Information Bottleneck (IB) method and the ICSI Speaker Diarization engine. The average Speaker Error of the Multiple Distant Microphone recordings from the NIST RT-06 evaluation set is shown and it can be seen that the novel initialization method using only MFCCs performs better than the ICSI baseline and the IB approach with MFCCs and Time Delay of Arrival (TDOA) features [Vijayasenan et al., 2008b].
In Table 8.2, the novel initialization method is compared to the ICSI baseline system

Engine	Speaker Error	Relative change
ICSI baseline MFCC	16.1 %	-
IDIAP IB MFCC	16.6 %	+3.1 %
IDIAP IB MFCC + TDOA	13.9 %	-13.7 %
Prosodic initialization	11.8 %	-26.7 %

Table 8.1: Comparison of the novel initialization method to the ICSI baseline system and the Information Bottleneck (IB) approach [Vijayasenan et al., 2008b] on the evaluation set of the NIST RT-06 (MDM condition). The IB method was tested with MFCCs only and with MFCCs in combination with delay features. The novel initialization method performs better than all the other configrations listed in the table.

using MFCCs in combination with prosodic features [Friedland et al., 2009b]. The novel initialization method has less parameters than the baseline system and makes use of the prosodic features only to perform the non-uniform initialization and the estimation of the number of initial clusters but not for the agglomerative clustering process. Nevertheless, the novel initialization method performs only slightly worse than the ICSI baseline system that uses MFCCs in combination with prosodic features and gives more than 30 percent relative improvement compared to the ICSI baseline system only using MFCCs on that particular data-set.
In the presented work, the prosodic features were only used to estimate appropriate values for the parameters and to do a novel initialization that is non-uniform. In the future, the ICSI Speaker Diarization MultiStream engine that successfully uses MFCCs in combination with prosodic features to do the agglomerative clustering (Table 8.2), has to be modified to use the non-uniform initialization method and the prosodic features extracted only on the speech frames. Instead of extracting the

Engine	Speaker Error	Relative change
ICSI baseline MFCC	15.00 %	-
ICSI baseline MFCC + prosodic features	9.50 %	-36.7 %
Prosodic initialization	10.10 %	-32.7 %

Table 8.2: Comparison of the novel initialization method to the ICSI baseline system on the evaluation set of the NIST RT-07 (SDM condition) [Friedland et al., 2009b]. The ICSI baseline system was tested with MFCCs only and with MFCCs in combination with prosodic features. The novel initialization method performs almost as well as the system that uses MFCCs in combination with prosodic features.

features on the whole recording, this extraction procedure only uses the speech segments what may result in more accurate feature values, because performing prosodic feature extraction also on non-speech regions may affect some features such as pitch negatively.
It was shown that the proposed method outperforms the ICSI baseline system (only using MFCCs) on five out of the six data sets that the method was tested on. The data-sets are all relatively similar in terms of the number of speakers they contain (all less than 16 speakers). It may be interesting to see what is happening if there are more than 16 speakers in a meeting for example. The novel method may be able to find the correct number of speakers, whereas the engine with the fixed number of initial clusters (16) will never be able to diarize a meeting with more than 16 speakers with the correct number of speakers because the agglomerative clustering approach does never split segments. Unfortunately, there was no time left and no adequate data-set found to perform such experiments.
From a speed/performance point of view, the extraction of the prosodic features and the clustering procedure to estimate the number of initial clusters to perform a non-uniform initialization adds about one times realtime to the processing time. On the other hand, the agglomerative clustering algorithm is sequentially merging clusters and statically choosing 16 initial clusters may be unnecessary, especially in the case of shorter segment durations. Thus, by choosing a more appropriate, smaller value for the number of initial clusters (as the presented parameter estimation is doing), less merging iterations need to be performed and the agglomerative clustering procedure could be sped up. Future work may optimize the code of the best novel initialization method and reduce the processing time.

Appendix A

Error rates per Meeting

In this appendix the Speaker Error per individual meeting of all the data-sets that are used during this work are listed. The Meeting ID *ALL* means the error over all the meetings concatenated together. This error rate may differ from the average error because not all the meetings contain the same amount of speech and the DER is time-based. In the tables, different methods are compared. As baseline system, the FastMatch implementation of the ICSI Diarization engine is chosen [Huang et al., 2007]. *Model1* is the method that assigns four Gaussians to every initial cluster and the number of initial clusters is determined by the linear regression based on the seconds per Gaussians (see Section 5.3). In Chapter 5, it is shown that the parameter *seconds per Gaussian* is important and can be estimated well by the linear regression model that was built. *Prosodic init* is the novel initialization method that is presented in Chapter 6. The number of initial clusters is estimated based on prosodic features, extracted with a window size of 1000 ms and the linear regression based on the seconds per Gaussian determines the number of Gaussians per initial cluster. The method *prosodic init*, has less parameters than the baseline implementation and has a better accuracy at the same time. It can be seen in the different tables, that the novel initialization method does not improve the result for every individual meeting, but the overall performance shows a 17.5 % relative improvement compared to the baseline system.

Meeting ID	FastMatch	model1	rel. diff.	prosodic init	rel. diff.
AMI_20041210-1052	6.90%	6.90%	0.0%	6.90%	0.0%
AMI_20050204-1206	6.30%	7.20%	+14.3%	6.00%	-4.8%
CMU_20050228-1615	7.30%	8.80%	+20.5%	5.60%	-23.3%
CMU_20050301-1415	3.30%	8.00%	+142.4%	7.60%	+130.3%
ICSI_20000807-1000	4.50%	11.10%	+146.7%	4.60%	+2.2%
ICSI_20010208-1430	5.20%	6.00%	+15.4%	10.80%	+107.7%
LDC_20011116-1400	1.80%	2.10%	+16.7%	4.70%	+161.1%
LDC_20011116-1500	11.90%	8.20%	-31.1%	7.10%	-40.3%
NIST_20030623-1409	7.30%	3.30%	-54.8%	2.70%	-63.0%
NIST_20030925-1517	17.10%	9.70%	-43.3%	8.90 %	-48.0%
VT_20050304-1300	3.00%	1.10%	-63.3%	2.60%	-13.3%
VT_20050318-1430	14.20%	16.80%	+18.3%	16.10%	+13.4%
ALL	6.90%	7.10%	+2.9%	6.60%	-4.4%

Table A.1: RT-06 development set, Speaker Error rates per meeting for different methods.

Meeting ID	FastMatch	model1	rel. diff.	prosodic init	rel. diff.
CMU_20050912-0900	17.80%	10.90%	-38.8%	10.80%	-39.3%
CMU_20050914-0900	7.40%	6.40%	-13.5%	6.20%	-16.2%
EDI_20050216-1051	19.80%	38.90%	+96.5%	21.20%	+7.1%
EDI_20050218-0900	20.50%	18.90%	-7.8%	20.80%	+1.5%
NIST_20051024-0930	11.00%	6.00%	-45.5%	11.60%	+5.5%
NIST_20051102-1323	4.50%	3.80%	-15.6%	9.30%	+106.7%
TNO_20041103-1130	21.00%	22.00%	+ 4.8%	15.50%	-26.2%
VT_20050623-1400	9.40%	7.40%	-21.3%	5.70%	-39.4%
VT_20051027-1400	38.70%	22.10%	-42.9%	3.70%	-90.4%
ALL	16.10%	14.70%	-8.7%	11.80%	-26.7%

Table A.2: RT-06 evaluation set (Multiple Distant Microphone condition), Speaker Error rates per meeting for different methods.

Meeting ID	FastMatch	model1	rel. diff.	prosodic init	rel. diff.
CMU_20050912-0900	12.50%	18.80%	+50.4%	9.40%	-24.80%
CMU_20050914-0900	8.80%	7.20%	-18.2%	7.20%	-18.2%
EDI_20050216-1051	27.20%	23.50%	-13.6%	45.30%	+66.6%
EDI_20050218-0900	18.90%	32.80%	+73.5%	29.30%	+55.0%
NIST_20051024-0930	3.30%	2.00%	-39.4%	5.60%	+69.7%
NIST_20051102-1323	4.10%	3.30%	-19.5%	4.40%	+7.3%
VT_20050623-1400	18.40%	11.00%	-40.2%	14.30%	-22.3%
VT_20051027-1400	3.80%	4.20%	+10.5%	3.50%	-7.9%
ALL	11.90%	12.80%	+7.6%	14.60%	+22.7%

Table A.3: RT-06 evaluation set (Single Distant Microphone condition), Speaker Error rates per meeting for different methods.

Meeting ID	FastMatch	model1	rel. diff.	prosodic init	rel. diff.
CMU_20061115-1030	12.50%	14.60%	+16.8%	12.30%	-1.6%
CMU_20061115-1530	9.70%	8.90%	-8.2%	7.70%	-20.6%
EDI_20061113-1500	23.00%	33.00%	+43.5%	14.40%	-46.1%
EDI_20061114-1500	12.80%	13.20%	+3.1%	13.50%	+5.5%
NIST_20051104-1515	2.70%	2.60%	-3.7%	2.90%	+7.4%
NIST_20060216-1347	2.00%	4.10%	+105.0%	3.40%	+70.0%
VT_20050408-1500	5.20%	2.30%	-55.8%	2.10%	-59.6%
VT_20050425-1000	11.20%	11.10%	-0.9%	11.70%	+4.5%
ALL	9.70%	11.0%	+13.4%	8.10%	-16.6%

Table A.4: RT-07 evaluation set (Multiple Distant Microphone condition), Speaker Error rates per meeting for different methods.

Meeting ID	FastMatch	model1	rel. diff.	prosodic init	rel. diff.
CMU_20061115-1030	16.10%	19.30%	+19.9%	12.3%	-23.6%
CMU_20061115-1530	14.60%	15.60%	+6.9%	9.30%	-36.3%
EDI_20061113-1500	32.20%	14.60%	-54.7%	12.20%	-62.1%
EDI_20061114-1500	26.50%	21.20%	-20.0%	12.60%	-52.5%
NIST_20051104-1515	3.00%	2.90%	-3.4%	2.60%	-13.3%
NIST_20060216-1347	3.10%	3.90%	+25.8%	2.90%	-6.5%
VT_20050408-1500	6.30%	4.60%	-27.0%	6.50%	+3.2%
VT_20050425-1000	22.40%	24.00%	+7.1%	24.00%	+7.1%
ALL	15.20%	13.00%	-14.5%	10.10%	-33.6%

Table A.5: RT-07 evaluation set (Single Distant Microphone condition), Speaker Error rates per meeting for different methods.

Meeting ID	FastMatch	model1	rel. diff.	prosodic init	rel. diff.
IS1000a	20.40 %	15.00 %	-26.5 %	15.70 %	-23.0 %
IS1001a	21.50 %	12.10 %	-43.72 %	6.10 %	-71.6 %
IS1001b	11.40 %	4.00 %	-64.9 %	11.80 %	+3.5 %
IS1001c	3.70 %	15.00 %	+30.4 %	21.00 %	+467.6 %
IS1003b	11.40 %	6.60 %	-42.11 %	7.90 %	-30.7 %
IS1003d	19.00 %	21.10 %	+11.1 %	20.60 %	+8.4 %
IS1006b	15.10 %	8.10 %	-46.36 %	8.80 %	-41.7 %
IS1006d	15.90 %	18.50 %	+16.35 %	15.40 %	-3.1 %
IS1008a	6.60 %	10.00 %	+51.52 %	2.10 %	+57.6 %
IS1008b	7.60 %	10.60 %	+39.47 %	8.90 %	-72.4 %
IS1008c	5.20 %	10.40 %	+100.0 %	3.80 %	-26.9 %
IS1008d	34.80 %	9.80 %	-71.84 %	7.80 %	-77.6 %
ALL	14.30 %	11.90 %	-16.8 %	11.10 %	-22.4 %

Table A.6: 12 chosen AMI meetings, Speaker Error rates per meeting for different methods.

Bibliography

Adami, A., Burget, L., Dupont, S., Garudadri, H., Grezl, F., Hermansky, H., Jain, P., Kajarekar, S., Morgan, N., and Sivadas, S. (2002). Qualcomm-icsi-ogi features for asr. In *Proc. ICSLP*, pages 4–7.

Ajmera, J. (2003). A robust speaker clustering algorithm. In *In Proceedings of IEEE Workshop on Automatic Speech Recognition Understanding*, pages 411–416.

Barras, C., Zhu, X., Meignier, S., and luc Gauvain, J. (2004). Improving speaker diarization. In *in Proc. Fall 2004 Rich Transcription Workshop (RT-04)*.

Bishop, C. M. (1995). Neural networks for pattern recognition.

Chen, S. S. and Gopalakrishnan, P. S. (1998). environment and channel change detection and clustering via the bayesian information criterion. pages 127–132.

Fiscus, J. and Ajot, J. (2007). Rt-07 speaker diarization results.

Fredouille, C. and Evans, N. (2008). The lia rt'07 speaker diarization system. pages 520–532.

Friedland, G., Hung, H., and Yeo, C. (2009a). Multi-modal speaker diarization of real-world meetings using compressed-domain video features.

Friedland, G., Vinyals, O., Huang, Y., and Muller, C. (2009b). Prosodic and other long-term features for speaker diarization.

HTK (2007). Htk 3.4. Available from: `http://htk.eng.cam.ac.uk/`.

Huang, Y., Vinyals, O., Friedl, G., Mller, C., Mirghafori, N., and Wooters, C. (2007). A fast-match approach for robust, faster than real-time speaker diarization. In *in ASRU*.

Huijbregts, M. A. H. (2008). *Segmentation, diarization and speech transcription : surprise data unraveled*. PhD thesis, Enschede. Available from: `http://doc.utwente.nl/60130/`.

Koh, C.-W. E., Sun, H., Nwe, T. L., Nguyen, T. H., Ma, B., Siong, C. E., Li, H., and Rahardja, S. (2007). Speaker diarization using direction of arrival estimate and acoustic feature information: The i2r-ntu submission for the nist rt 2007 evaluation. In Stiefelhagen, R., Bowers, R., and Fiscus, J. G., editors, *CLEAR*, volume 4625 of *Lecture Notes in Computer Science*, pages 484–496. Springer. Available from: `http://dblp.uni-trier.de/db/conf/clear/clear2007.html#KohSNNMCLR07`.

Leeuwen, D. A. (2005). The tno speaker diarization system for nist rt05s meeting data.

Leeuwen, D. A. and Konečný, M. (2008). Progress in the amida speaker diarization system for meeting data. pages 475–483.

Luque, J., Anguera, X., Temko, A., and Hernando, J. (2008). Speaker diarization for conference room: The upc rt07s evaluation system. pages 543–553.

Marin, M. J., Mengerson, K., and Robert, C. P. (2005). Bayesian modelling and inference on mixtures of distributions. *Handbook of Statistics*, pages 15840–15845.

Miró, A. (2006). Beamformit 2.0. Available from: `http://www.icsi.berkeley.edu/~xanguera/beamformit/`.

Miró, A., (upc), A. D. F. J. H. P., and (icsi), D. C. W. (2006). *PhD Thesis Robust Speaker Diarization for*. PhD thesis.

NIST (2009). Uem file format. Available from: `http://nist.gov/speech/tests/rt/2009/docs/rt09-meeting-eval-plan-v1.pdf`.

Praat (2009). Praat5032. Available from: `http://www.fon.hum.uva.nl/praat/`.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. Available from: `http://dx.doi.org/10.1109/5.18626`.

Tishby, N., Pereira, F. C., and Bialek, W. (1999). The information bottleneck method. pages 368–377.

Vijayasenan, D., Valente, F., and Bourlard, H. (2008a). An information theoretic approach to speaker diarization of meeting data. IDIAP-RR 58, IDIAP.

Vijayasenan, D., Valente, F., and Bourlard, H. (2008b). Integration of tdoa features in information bottleneck framework for fast speaker diarization. In *Interspeech 2008*. IDIAP-RR 08-26.

Weka (2008). Weka 3.4.13. Available from: `http://www.cs.waikato.ac.nz/ml/weka/`.

Wooters, C. and Huijbregts, M. (2007). The icsi rt07s speaker diarization system. In Stiefelhagen, R., Bowers, R., and Fiscus, J. G., editors, *CLEAR*, volume 4625 of *Lecture Notes in Computer Science*, pages 509–519. Springer. AMIDA-59.

Zhu, X., Barras, C., Lamel, L., and Gauvain, J.-L. (2008). Multi-stage speaker diarization for conference and lecture meetings. pages 533–542.

Glossary

BIC The Bayesian Information Criterion is a common method of selecting between competing models. It imposes a trade-off between model quality and model complexity [Chen and Gopalakrishnan, 1998] and it is widely used in the domain of Speaker Diarization. The common approach makes use of a penalty term which is weighted by the so-called parameter λ that needs to be tuned.

DER The Diarization Error Rate is the primary metric for the evaluation (NIST RT) of the Speaker Diarization task and is defined as follows:

$$\frac{\sum_{all\ \ segments}\{dur(seg)\cdot(max(N_{ref}(seg),N_{sys}(seg))-N_{correct}(seg))\}}{\sum_{all\ \ segments} dur(seg)\cdot N_{ref}(seg)}$$

where the speech data file is divided into contiguous segments at all speaker change points and where, for each segment seg:
$dur(seg)$ = the duration of seg
$N_{ref}(seg)$ = the # of reference speakers speaking in seg
$N_{sys}(seg)$ = the # of system speakers speaking in seg
$N_{correct}(seg)$ = the # of reference speakers speaking in seg for whom their matching (mapped) system speakers are also speaking in seg.

EM Expectation-maximization is an algorithm to estimate the mixture parameters of the maximum likelihood solution, see [Bishop, 1995] page 65.

GMM The term Gaussian Mixture Model is often used in this work and abbreviated as GMM. For more information about Gaussian Mixture Models, see [Marin et al., 2005].

HMM A Hidden Markov Model (HMM) is a statistical model. The modeled system is assumed to be a Markov process with unknown parameters. For more information about Hidden Markov Models, see [Rabiner, 1989].

IB Information Bottleneck, an information theoretic framework, see [Tishby et al., 1999].

MDM Multiple Distant Microphones, an evaluation condition used during the NIST evaluations [1].

NIST National Institute of Standards and Technology[1].

PDF Probability density function, see [Bishop, 1995] page 21.

RT The National Institute of Standards and Technology (NIST) defines Rich Transcription (RT) to be a fusion of speech-to-text technology and metadata extraction technologies which will provide the basis for the generation of more usable transcriptions of human-human speech in meetings for both humans and machines. NIST organizes evaluations in this domain. In this work, references to this evaluations are indexed with the corresponding year they were hold. NIST RT-07 for example means the NIST Rich Transcription evaluation that was hold in spring of the year 2007. The next evaluation will take place in Spring 2009[1].

RTTM File format: Rich Transcription Time Marked[1].

SAD Speech Activity detection means the process of dividing an audio track into segments with presence or absence of human speech. This step is often executed at the beginning of the Speaker diarization task.

SDM Single Distant Microphones, an evaluation condition used during the NIST evaluations [1].

UEM File format: Un-partitioned evaluation map[1].

Viterbi The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states (the so-called Viterbi path) that results in a sequence of observed events.

[1] http://www.nist.gov/speech/tests/rt/2009/index.html