The Blame Game: Performance Analysis of Speaker Diarization System Components

Marijn Huijbregts[1,2]*, Chuck Wooters*[2]

[1]Department of Electrical Engineering, Mathematics and Computer Science,
University of Twente, Enschede, The Netherlands
[2]International Computer Science Institute,
1947 Center Street, Suite 600, Berkeley, CA 94704, USA
marijn.huijbregts@ewi.utwente.nl, wooters@icsi.berkeley.edu

Abstract

In this paper we discuss the performance analysis of a speaker diarization system similar to the system that was submitted by ICSI at the NIST RT06s evaluation benchmark. The analysis that is based on a series of oracle experiments, provides a good understanding of the performance of each system component on a test set of twelve conference meetings used in previous NIST benchmarks. Our analysis shows that the speech activity detection component contributes most to the total diarization error rate (23%). The lack of ability to model overlapping speech is also a large source of errors (22%) followed by the component that creates the initial system models (15%).
Index Terms: speaker diarization, rich transcription

1. Introduction

The goal of speaker diarization is to automatically segment an audio recording into speaker homogeneous regions. Although the identity of each speaker is not known and even the number of speakers is unknown, a diarization system should be able to anonymously label each speaker in the recording and answer the question: 'Who spoke when?' [1].

Since 2002 NIST has organized evaluations of speaker diarization technology. Since 2004 these evaluations are also performed on the meeting domain [2]. ICSI has successfully participated in these benchmarks with a system based on a Hidden Markov Model architecture and Gaussian Mixture Models that are trained using only the speech in the data under evaluation. Several system improvements were introduced in resent years ([3]), but the system framework initially proposed in [4], is still being used.

In [5] various features of audio data like the number of speakers and speaker turns, were analyzed in order to get a better understanding of what makes an audio recording hard to diarize correctly. In this research, instead of looking at data characteristics, we will investigate the behaviour of each component in our system so that we can determine which parts of the system are performing well and which parts are responsible for most of the Diarization Error Rate (DER). The analysis approach consists of a series of oracle experiments. We start by replacing as many components as possible with components that know 'the truth' and that will not make any mistakes. By placing the real components back into the system one at a time, we can measure the performance of each component.

In the next section, a short description of the speaker diarization system that is being inspected will be given. In section 3 the oracle experiments will be described. In section 4 the results of these experiments will be summarized. In section 5 these results will be interpreted and discussed.

2. System Description

The speaker diarization system discussed in this paper is implemented as part of the SHOUT toolkit (SHOUT is a Dutch acronym for: 'speech recognition research University of Twente'). Its architecture is similar to that of the ICSI RT07s speaker diarization submission. The main difference is that this system does not use a second Gaussian Mixture Model (GMM) for delay-sum features (see [3]). In section 2.1 the feature extraction will be described. In section 2.2 the system topology will be discussed followed by a description of the algorithm used to obtain the final topology.

2.1. Feature Extraction

The conference meetings on which the diarization system will be evaluated, were recorded with multiple microphones. The audio signal of each farfield microphone is first passed through a Wiener filter for noise reduction and then the channels are combined using beam-forming software[1] ([3]).

From the resulting 16Khz audio file, Mel Frequency Cepstral Coefficients (MFCC) are extracted. The feature vectors are calculated using 32ms Hamming windows that are shifted 10ms at a time. Twenty-four melscale filters are used to calculate vectors containing the first nineteen cepstral coefficients.

2.2. Topology

The system is based on the use of Hidden Markov Models (HMM) with GMMs as probability density functions. Figure 1 shows the topology. The HMM states drawn horizontally all share a single GMM. In an ideal situation, each GMM is trained on all the speech of one unique speaker. The speaker segmentation, the final system result, is found by performing a Viterbi alignment of all audio that contains speech. All audio that is processed by the same string of states during this alignment is grouped together as speech from one speaker. By using a string of states to represent each speaker (instead of a single state), a minimum duration of each speech segment is guaranteed.

Obviously, the initial state of the system will not automatically be the preferred situation (where each speaker is represented by exactly one string of states). Initially too many HMM

[1]http://www.icsi.berkeley.edu/~xanguera/beamformit

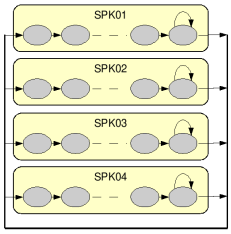

Figure 1: The HHM topology of the speaker diarization system. Each string of states represents a unique speaker. Only the final state of each string has a loop transition and all states in a string share the same GMM.

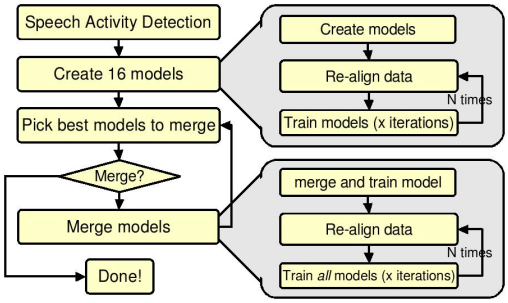

Figure 2: A schematic representation of the speaker diarization algorithm. In order to evaluate the five steps of this algorithm, each step can be replaced by an oracle component. The second and fifth step each consist of a number of training and realignment iterations.

states are created. The number of states is then iteratively decreased and the GMMs are slowly trained on speech from a single speaker until the correct number of GMMs is reached. The following algorithm is used to do this.

2.3. Algorithm

In order to obtain the optimal topology, an algorithm with five steps is executed. An overview of these steps is drawn in figure 2.

First, Speech Activity Detection (SAD) is performed. This step should filter all non-speech out of the audio signal. The speech segments are then passed to the speaker diarization system for initialization.

The system will be initialized with a large number of models (strings of HMM states). The number of models should be significantly higher than the assumed maximum number of speakers in the audio file. For the RT07s benchmark we used 16 initial clusters for each meeting. In order to create the initial sixteen speaker models, the available speech data is cut up into small pieces and these pieces are randomly divided in a number of bins. Each bin is used to train one of the GMMs. Each GMM is initialized with one gaussian and the number of gaussians is increased by splitting the gaussian with the biggest weight after each training iteration, until the GMM contains five gaussians. Although the GMMs are trained using multiple speakers, in general one speaker will fit the GMM a little bit better than the other speakers. Therefore, when a Viterbi alignment is performed, this GMM will be assigned more speech from this speaker. The data will be re-aligned a number of times and after each iteration, the GMMs are re-trained. Each iteration the model will fit the dominant speaker better than before. The result of this step is a group of models that are all trained with as much data as possible of one dominant speaker and as little data as possible of the other speakers. In the remaining steps, the models that are trained on the same dominant speaker will need to be merged.

In the third step it is determined which two models are most likely trained on the same speaker. This is done by calculating the local Bayesian Information Criterion (BIC) score for each combination of two models. For this BIC comparison, a new model θ is trained containing the sum of the number of gaussians of the two original models θ_a and θ_b. This *merged* model is trained on the training data of the original two models. If the two models are trained on data of one single speaker, this merged model must be able to replace the two models without decreasing system performance. The following formula is used to calculate this change in system performance. D_a is the data used to train model θ_a, D_b to train θ_b and D to train model θ.

$$BIC(\theta_a, \theta_b) = logP(D|\theta) - logP(D_a|\theta_a) - logP(D_b|\theta_b) \quad (1)$$

Note that no scaling parameter is needed in this equation, because the total number of gaussians, the system complexity, will remain unchanged after replacing θ_a and θ_b by θ. If the BIC score is positive, θ_a and θ_b are considered to be trained on data of the same speaker. The higher the BIC score, the more the two models were similar. Therefore, the cluster pair with the highest BIC score will be chosen as the candidate for merging and in the fourth step, the decision to merge the candidate cluster pair is positive if the BIC score is bigger than zero and negative if the BIC score is negative.

In this last case the final Viterbi alignment will be performed and the algorithm is finished. But if the BIC score is positive, in the fifth step, the two merge candidates are replaced in the HMM by the merged model. The data is re-aligned over the models and the models are re-trained with the new data. After this, a new merging iteration is started.

2.4. Parameters

The speaker diarization system is designed to have no system parameters that need to be tuned on external data, making it robust for changes in audio conditions or application domain. Because the GMM models are trained on the data under evaluation, no models created on a training set are needed. Also, because BIC is used with preservation of system complexity, no scaling parameter is needed. Unfortunately, a number of system parameters still exist, although these parameters do not seem sensitive for changes in audio conditions.

For this evaluation we have chosen the system parameters identical to the parameters used in RT06s ([3]). The number of initial clusters is set to 16 and each GMM is initialized with 5 gaussians. The minimum segment duration (the number of HMM states) is set to 250 (2.5 seconds). No state transition penalties are used.

3. The Blame Game

On a test set consisting of twelve conference meetings from previous NIST Rich Transcription benchmark evaluations, the diarization error rate (DER) of the system was 11.76% (see table 1). The DER is calculated using the NIST metrics. The

Meeting ID	% SAD error	% Spkr error	DER
AMI_20041210-1052	1.60	7.10	8.71
AMI_20050204-1206	4.80	4.50	9.27
CMU_20050228-1615	10.50	4.80	15.28
CMU_20050301-1415	5.40	1.90	7.27
ICSI_20000807-1000	5.00	4.00	9.04
ICSI_20010208-1430	4.80	12.80	17.61
LDC_20011116-1400	5.10	4.40	9.45
LDC_20011116-1500	7.00	7.90	14.93
NIST_20030623-1409	1.70	2.70	4.31
NIST_20030925-1517	13.40	12.40	25.79
VT_20050304-1300	1.60	3.10	4.72
VT_20050318-1430	7.50	16.40	23.87
Overall	5.30	6.40	11.76

Table 1: The results of the diarization system on our test set of twelve conference meetings. The Diarization Error Rate (DER) is the sum of the SAD error and the error due to classifying speech as the wrong speaker (spkr error).

total error is the sum of two rates: the speech activity detection error and the speaker classification error. The SAD error is the percentage of speech and non-speech that is misclassified. The speaker classification error is the error due to misclassifying speakers [2]. In order to determine which components are responsible for which part of this error, a series of experiments was performed. These experiments typically are oracle experiments. For each experiment, part of the system is replaced with an oracle system: a system that does not make mistakes.

3.1. Reference Transcripts

In order to perform the oracle experiments, the reference transcripts were used in three different ways. In the first experiments of the series, the transcripts were used as input for the diarization system. One way of doing this is to replace the SAD output with the transcription. In this case the IDs of all speakers in the transcripts were replaced with one ID ('speech') and all overlap regions were replaced by single regions. Another way we used the reference as input is to replace the initial clustering with a perfect clustering obtained from the reference transcript.

The reference transcripts were also used to make merging decisions. Instead of performing BIC, the oracle merge component will score a segmentation with the NIST scoring tools[2] and determine for each cluster which speaker it represents most (which speaker was classified by that cluster the longest period of time). The purity of each cluster is then calculated as follows: The purity of cluster A is the time that the representing speaker was classified as A divided by the total amount of time that was classified as A. The oracle merge component will then decide to merge the two clusters with the same representing speaker that have the highest purity.

Third, the system was modified so that an intermediate hypothesis segmentation file could be printed after each merging iteration. The reference transcript was used for scoring these segmentations for monitoring purposes.

3.2. Oracle Experiments

In total, six oracle experiments are conducted. In the first experiment, all algorithm steps (see section 2.3) are replaced by oracle components and at each following experiment one of the components is placed back into the system.

3.2.1. Perfect Topology

If the algorithm would do a perfect job, the HMM would contain exactly one model per speaker and each model is trained on all the available speech of its speaker. Even if the algorithm did not make a single mistake and this perfect topology was created, we do not expect the system to have a perfect diarization score because the system is not able to model overlapped speech and because the models, with their limited number of gaussians, might not be able to classify all speech perfectly. In order to test the system on these limitations, in the first experiment the reference transcription is used instead of the SAD component. For each speaker in the reference, one model is trained on all its speech segments. The total number of gaussians in the system is the same as normal (80) and they are divided over the clusters based on the amount of speech that is available for each speaker. After the models are created, a Viterbi pass is performed to find the final system result.

3.2.2. Speech Activity Detection

In the second experiment, the real SAD component is placed back into the system. The models are still trained directly on the speech from the reference transcription, but the final alignment is performed with the actual SAD segmentation. The increase of error rate compared to the previous experiment is the amount of the total system error that can be blamed on the SAD component.

3.2.3. Merging Algorithm

We now want to test what the influence of using the actual merging algorithm is on the final result. For this we use the reference transcription to create sixteen initial models. Each model is created with speech of only one speaker, but because we now need sixteen models, the speech of each speaker is cut up in pieces so that multiple models can be trained on each speaker. The data is divided so that each cluster is trained on an average amount of data (a person that spoke a lot in a meeting, will have a high number of initial models). The normal model initialization and merging procedure will be used, but the decisions about which models to merge and when to stop is performed by the oracle components (as described in section 3.1). The DER scored on this experiment subtracted by the error of the previous experiment will reveal the error that is introduced by the procedure of creating the final models by merging the smaller initial models together.

3.2.4. Model Initialization

Instead of creating the initial models with use of the reference transcript, in this experiment the initial models are created normally by dividing the speech data randomly. The merging and stop decisions are still performed by the oracle components though. Therefore the increase of error can be assigned to the shortcomings of the systems model initialization method.

3.2.5. Merge Candidate Selection

The oracle merge candidate selection component is now replaced by the original candidate selection based on BIC. The increase of DER reveals the shortcomings of this selection method.

[2] `http://www.nist.gov/speech/tools`

Oracle experiment	SAD (%)	DER (%)
Perfect Topology	2.60	4.18
Speech Activity Detection	5.30	6.87
Merging Algorithm	5.30	7.36
Model Initialization	5.30	9.18
Merge Candidate Selection	5.30	10.40
Stop Criterion	5.30	11.76

Table 2: The SAD and DER errors of the six oracle experiments. The experiments are named after the description titles in section 3.2

Test description	DER (%)	Relative
Overlapping speech	2.60	22.11
Speech Activity Detection	2.70	22.96
Modeling/alignment	1.57	13.35
Merging algorithm	0.49	4.17
Non-perfect initial clusters	1.82	15.48
Combining wrong models	1.22	10.37
Stop clustering too early/late	1.36	11.56
System DER (sum of components)	11.76	100.00

Table 3: The contribution of each system step to the overall DER.

3.2.6. Stop Criterion

The final oracle component that is replaced by its original is the component that decides when to stop merging. The difference between the real system DER and the previous experiment will tell us the error gained because of incorrect stop decisions.

4. Experiment Results

We performed the six experiments described in section 3.2 on our test set of twelve conference meetings. The results of these experiments are listed in table 2. In the first experiment, the entire algorithm to create the HMM topology is bypassed and the models were created directly. At each following experiment, one step of the algorithm is placed back into the system. Assuming that the components are mostly performing independent of each other (see section 5), at each step, the increase in DER is a good indication of the contribution to the total error of the component placed back.

Although in the first experiment, the reference SAD was used, the SAD error was still 2.60%. This error is due to the inabillity of the system to model overlapped speech. The DER in this experiment was 4.18%. That means that although the merging algorithm was bypassed, the modelling approach is not perfect and is responsible for 1.58% DER. In the second experiment, where the real SAD component is used, the SAD error increased to 5.3%. Part of this error is because of overlapped speech (2.6%) but 2.7% is due to errors in the SAD component. The increase in DER in the third to the sixth experiments can be assigned to the use of the merging algorithm (0.49%), initializing the clusters (1.82%), performing BIC to combine models (1.22%) and determining when to stop merging (1.36%). Table 3 summarizes the contribution to the DER of each component.

As can be seen in table 3 the three factors that contribute most to the total DER are the lack of being able to model overlapped speech, the speech activity detection itself and the initialization of the sixteen clusters.

5. Discussion and Conclusions

In this paper we described the architecture of a speaker diarization system that trains its GMMs only on the data under evaluation according to a five step algorithm. The system was analyzed by performing a series of oracle experiments.

During this analysis we asumed that the performance of each component is mostly independent of the performance of others. Unfortunately though, changing one component *is* likely to have an impact on other components so assigning the entire increase in error to one component like we did in section 4 might sometimes give a slightly distorted picture. A way to investigate the dependencies between components is to test each component with input of varying quality. For example, input of the oracle component could be used.

It should be noted that the results of this analysis are partly dependent on the evaluation data and that a high contribution to the error of a certain component does not necessarily mean that this component can be improved the most. Also, these results apply to our diarization system and might be different for other systems. Given these caveats, this analysis has helped us to better understand the system behaviour and any changes we applied to it.

One of the components that contributed most to the total DER is the speech activity component. At RT07s we will introduce a new SAD component that we have developed after this analysis. The SAD error of the new component on the evaluation set used in this paper is 4.5%, an improvement of 0.8% absolute. Currently we are investigating alternative initialization approaches and methods to detect overlapping speech.

6. Acknowledgements

The work reported here was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multiparty Interaction, FP6-506811) and by the bsik-program MultimediaN which is funded by the Dutch government (`http://www.multimedian.nl`).

7. References

[1] D. Reynolds and P. Torres-Carrasquillo, "Approaches and applications of audio diarization," in *Proceedings of ICASSP*, March 2005, pp. 953–956.

[2] J. Fiscus, J. Ajot, M. Michel, and J. Garofolo, "The rich transcription 2006 spring meeting recognition evaluation," in *NIST Rich Transcription 2006 Spring Meeting Recognition Evaluation, RT06s, Washington DC, USA*, ser. Lecture Notes in Computer Science, vol. 4299. Berlin: Springer Verlag, October 2007, pp. 309–322.

[3] X. Anguera, C. Wooters, and J. Pardo, "Robust speaker diarization for meetings: Icsi rt06s evaluation system," in *NIST Rich Transcription 2006 Spring Meeting Recognition Evaluation, RT06s, Washington DC, USA*, ser. Lecture Notes in Computer Science, vol. 4299. Berlin: Springer Verlag, October 2007.

[4] J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *IEEE ASRU Workshop*, 2003.

[5] N. Mirghafori and C. Wooters, "Nuts and flakes: A study of data characteristics in speaker diarization," in *Proceedings of ICASSP*, 2006.