The ICSI RT07s Speaker Diarization System

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

[1] International Computer Science Institute, Berkeley CA 94704, USA,
[2] University of Twente
Department of Electrical Engineering, Mathematics and Computer Science,
Enschede, The Netherlands
{wooters,marijn}@icsi.berkeley.edu

Abstract. The goal of speaker diarization is to segment an audio recording into speaker-homogeneous regions [1]. Typically, this segmentation must be performed with little knowledge of the characteristics of the audio or of the talkers in the recording. For example, we may know the source and date of the audio recording (e.g. CNN Nightly News or a meeting that was held at CMU), but we typically do not know how many speakers occur in the recording, how many male vs. female talkers occur, or whether there are commercials, music, or other noises, etc.
In this paper, we present the ICSI speaker diarization system. This system was used in the 2007 National Institute of Standards and Technology (NIST) Rich Transcription evaluation. The ICSI system automatically performs both speaker segmentation and clustering without any prior knowledge of the identities or the number of speakers. Our system uses "standard" speech processing components and techniques such as HMMs, agglomerative clustering, and the Bayesian Information Criterion. However, we have developed the system with an eye towards robustness and ease of portability. Thus we have avoided the use of any sort of model that requires training on "outside" data and we have attempted to develop algorithms that require as little tuning as possible.

1 Introduction

The goal of speaker diarization is to segment an audio recording into speaker-homogeneous regions. This task is sometimes referred to as the "Who Spoke When" task. Knowing when each speaker is speaking is useful as a pre-processing step in speech-to-text (STT) systems to improve the quality of the output. Such pre-processing may include vocal tract length normalization (VTLN) and/or speaker adaptation. Automatic speaker segmentation may also be useful in information retrieval and as part of the indexing information of audio archives.

For the past three years, the US National Institute of Standards and Technology (NIST) has conducted competitive evaluations of speaker diarization systems using recordings from multi-party meetings. For these evaluations, the speaker diarization task must be performed with little knowledge of the characteristics of the audio or of the talkers in the recording. Within the meeting domain, there are several conditions on which diarization systems are evaluated. The

primary evaluation condition allows the use of audio recorded from multiple distant microphones. As an optional task, NIST also evaluates the performance of diarization systems when the audio input comes from just a single (distant) microphone. The performance of diarization systems on the multiple distant microphone task is typically better than on the single distant microphone task due to the extra information provided by the additional microphones.

One of the most commonly used techniques for performing speaker diarization is agglomerative clustering, where a large number of initial models are merged pair-wise, until the system arrives at a single model per speaker. Techniques such as agglomerative clustering often call for the use of "tunable" parameters such as: the number of initial models, the number of Gaussian mixtures per model, or the penalty factor used in the Bayesian Information Criterion (BIC) [2]. The choice of the values for these parameters can be vital to the performance of the clustering system. Typically, system designers choose the values for the parameters empirically based on training and development data. It is important that this data be as similar as possible to the data on which the system will ultimately be tested in order to ensure robust behavior.

In this paper, we present the ICSI speaker diarization system used in the NIST RT07s evaluations. The system we present is based on agglomerative clustering and automatically deduces the number of speakers in a recording, along with the information about where each speaker is speaking. The algorithm runs iteratively, alternating model alignment with model merging. The algorithm we use for model merging is a modification of BIC in which we keep the number of parameters between the two BIC hypotheses constant. An important property of this modification of BIC is that it allows us to eliminate the BIC penalty term, thus eliminating one of the parameters that must be tuned.

In section 2, we present an overview of the speaker diarization system that we used for the 2007 evaluation. In section 3 we describe several experiments we ran after the evaluation to examine the behavior of the system in more detail. Finally, we end with some conclusions and future work.

2 System Description

2.1 Front-end Acoustic Processing

The acoustic processing consists of three steps. First, Wiener filtering [3] is performed on each available audio channel. The goal of the Wiener filtering is to remove any "corrupting" noise from the signal. The noise is assumed to be additive and of a stochastic nature. The implementation of the Wiener filtering we use was taken from the noise reduction algorithm developed for the Aurora 2 front-end proposed by ICSI, OGI, and Qualcomm [4]. The algorithm performs Wiener filtering with typical engineering modifications, such as a noise over-estimation factor, smoothing of the filter response, and a spectral floor. We modified the algorithm to use a single noise spectral estimate for each meeting waveform. This was calculated over all the frames judged to be non-speech by the voice-activity detection component of the Qualcomm-ICSI-OGI front end.

After Wiener filtering, if multiple audio channels (i.e. recordings from multiple microphones) are available, a single "enhanced" channel is created by running delay and sum beamforming on the separate channels. The beamforming was done using the BeamformIt 2.0 toolkit[3] with a 500 msec analysis window stepped at a 250 msec frame rate. Finally, feature extraction is performed on the resulting beamformed channel.

Our system uses two types of acoustic features. The first nineteen Mel Frequency Cepstrum Coefficients (MFCC), created using the HTK toolkit[4], form our standard feature type. These features are created at a 10 msec frame rate with a 30 msec analysis window. The second type of feature we use is only used when multiple audio channels are available (the MDM condition). In this case, we use the BeamformIt tool to calculate the delay values between the different audio channels. When using BeamformIt to produce these delay features, we use a 500 msec analysis but it is stepped at the same frame rate (10 msec) as was used for the MFCC features. The delay factors are then added to the system as a second feature stream.

2.2 Speech/Non-speech Detection

One of the improvements we made this year was the creation of a new speech/non-speech detector. The detector we used last year consisted of two stages. It first selected those regions in the audio with high and low energy levels and then in the second stage it trained dedicated speech models on the high energy regions and silence models on the low energy levels. The major advantage of this approach is that it does not use models trained on outside data making it robust to changes in audio conditions. The drawback of using energy however is that it is not possible to use this approach when the audio contains fragments with high energy levels that are non-speech. For another task [5], we developed a new speech/non-speech detector inspired by last year's system. This new system is better able to detect audible non-speech.

The new speech/non-speech detector consists of three steps. First, as in last years system, an initial guess is made about which regions in the audio are speech, silence or non-speech sounds. Only the regions that are classified with a high confidence score are labeled. To create these three regions, an initial segmentation is created with an HMM that contains a speech and a silence GMM that was trained on broadcast news data. The silence region is then split into two classes: regions with low energy and regions with high energy and high zero-crossing rates. From the data in each of these two classes a new GMM is trained. The GMM trained on the low energy data contains 7 gaussians, and the GMM trained on the high energy, high zero-crossing rate data contains 18 gaussians (once fully trained.) For the speech regions, a third GMM is trained with 24 gaussians. The gaussians of all three models are built up iteratively, and during this process the audio is re-segmented a number of times [6].

[3] Available at: http://www.icsi.berkeley.edu/x̃anguera/beamformit

[4] Available at http://htk.eng.cam.ac.uk/

In the second step of the speech/non-speech detector, models are trained from the data in the three regions defined by the first step, and we label these regions: "speech", "silence" and "non-speech sound". We always assume that a recording has all three types of regions. However, if an audio recording does not contain any non-speech sounds, it is possible that the "sound" model will end up containing "speech" data. Therefore, in the third step, the system checks to see if the "sound" and "speech" models are similar. To test for similarity, a new model is trained on the combined speech and sound data, and BIC is used to test whether it is better to model all of the data with one combined model or two separate models (similar to what we do during the diarization process). If the BIC score is positive, the sound model is discarded and a new speech model is trained using all of the speech and sound data.

For all of these steps, we use feature vectors with 12 MFCC components, zero-crossing, deltas and delta-deltas. The underlying system uses a Hidden Markov Model with two (or three) "strings" of states in parallel (in order to enforce a minimum duration for each segment). Each string shares a single Gaussian Mixture Model (GMM) as its probability density function that represents one of the three classes. The segmentation into the two (or three) classes is found by performing a Viterbi search on the data using this HMM.

2.3 Cluster Modeling

The Probability Density Function (PDF) used in our diarization system is modeled with a Gaussian Mixture Model (GMM). If multiple audio channels are available, each of the two audio streams (MFCC and delays) are modeled using separate GMMs, and the overall PDF is modeled as a weighted combination of these two GMMs. The weights of the two streams are initially set to fixed values (0.65 and 0.35). Then, during the merging process, the weights are adapted using the algorithm introduced in [7]. This approach makes it possible to automatically find appropriate weights for the two streams during the diarization process and eliminates the need to tune the weights on a development set.

2.4 Diarization Algorithm

As explained in [8] and [9], the speaker clustering system is based on an agglomerative clustering technique. It initially splits the data into K clusters (where K should be greater than the number of true speakers), and then iteratively merges the clusters (according to metric based on ΔBIC) until a stopping criterion is met. Our clustering algorithm models the acoustic data using an ergodic hidden Markov model (HMM), where the initial number of states is equal to the initial number of clusters (K). Upon completion of the algorithm's execution, each remaining state is taken to represent a different speaker. Each state in the HMM contains a set of MD sub-states, imposing a minimum duration on the model (we use $MD = 2.5$ seconds). Also, each one of the sub-states shares a single probability density function (PDF).

The following outlines the clustering algorithm step-by-step.

1. Run front-end acoustic processing.
2. Run speech/non-speech detection.
3. Extract acoustic features from the data and remove non-speech frames.
4. Create models for the K initial clusters via linear initialization.
5. Perform several iterations of segmentation and training to refine the initial models.
6. Perform iterative merging and retraining as follows:
   (a) Run a Viterbi decode to re-segment the data.
   (b) Retrain the models using the Expectation-Maximization (EM) algorithm and the segmentation from step (a).
   (c) Select the cluster pair with the largest merge score (based on ΔBIC) that is > 0.0.
   (d) If no such pair of clusters is found, stop and output the current clustering.
   (e) Merge the pair of clusters found in step (c). The models for the individual clusters in the pair are replaced by a single, combined model.
   (f) Go to step (a).

For our stopping criteria, we use ΔBIC, a variation of the commonly used Bayesian Information Criterion (BIC) [2]. ΔBIC compares two possible hypotheses: 1) a model in which the two clusters belong to the same speaker or 2) a model in which to two clusters represent different speakers. The variation used was introduced by Ajmera et al. [9], [10], and consists of the elimination of the tunable parameter (λ) by ensuring that, for any given ΔBIC comparison, the difference between the number of free parameters in the two hypotheses is zero.

3 Experiments and Results

3.1 Data

All of the experiments reported here were conducted using data distributed by NIST as part of the Rich Transcription 2004, 2005, 2006 and 2007 meeting recognition evaluations [11]. This data consists of excerpts from multi-party meetings collected at eight different sites. From each meeting, an excerpt (chosen by NIST) of 10 to 12 minutes is used.

RT07s Development Data Table 3.1 lists the names of the 21 meetings we used for our development testing for this year's evaluation. Because of the "flakiness" of the diarization error rate (DER) (see 3.2 for an explanation of DER) that we have observed in previous work [12], we believe that it is important to use as many meetings as possible for development. This helps to "smooth" diarization error rates by preventing large variations in the scores of one or two meetings from influencing the overall DER.

We performed all of our development work on the Multiple Distant Microphone (MDM) condition.

ICSI_20000807-1000	ICSI_20010208-1430
LDC_20011116-1400	LDC_20011116-1500
NIST_20030623-1409	NIST_20030925-1517
AMI_20041210-1052	AMI_20050204-1206
CMU_20050228-1615	CMU_20050301-1415
VT_20050304-1300	VT_20050318-1430
CMU_20050912-0900	CMU_20050914-0900
EDI_20050216-1051	EDI_20050218-0900
NIST_20051024-0930	NIST_20051102-1323
TNO_20041103-1130	VT_20050623-1400
VT_20051027-1400	

Table 1. The names of the 21 meetings used for development.

CMU_20061115-1030	CMU_20061115-1530
EDI_20061113-1500	EDI_20061114-1500
NIST_20051104-1515	NIST_20060216-1347
VT_20050408-1500	VT_20050425-1000

Table 2. The names of the eight RT07s evaluation meetings.

RT07s Evaluation Data Table 3.1 list the eight meetings that were chosen by NIST for this year's evaluation.

3.2 Error Metric

The metric used to evaluate the performance of the system is the same as is used in the NIST RT evaluations and is called Diarization Error Rate (DER). It is computed by first finding an optimal one-to-one mapping of reference speaker ID to system output ID and then obtaining the error as the percentage of time that the system assigns the wrong speaker label. All results presented here use the official NIST DER metric.

3.3 Experiments

Speech/Non-speech Detection This speech/non-speech system described in Section 2.2 outperformed last year's speech/non-speech system on our development set. Although typically in the meeting domain the number of non-speech sounds is negligible, in two of the twenty one meetings of our development set, the system classified part of the audio as non-speech sounds (paper shuffling and doors slamming). In the other meetings, (including all of the meetings in the test set) the BIC score was always positive and each sound model was discarded.

Table 3 contains the results of last year's system and our new system on the test set. The first two rows show the results scored only for speech activity detection. The new system has a slightly lower false alarm rate. The last two rows of table 3 show the results of our current diarization system using either the

speech/non-speech segmentation of the RT06 detector or the RT07 detector. On this data, the new detector has a lower false alarm rate. Most of the performance gain though is a result of the reduction in speaker error (diarization). This is partly explainable by the fact that we do not smooth the speech/non-speech data before diarization (see the next experiment). We surmise that the remainder of the gain is due to the reduced number of false alarms. We believe that this helps to make the data used to train the GMMs "cleaner", resulting in better models.

System	% missed speech	% false alarm	% SAD	% Spkr	% DER
RT06 (only SAD)	1.10	2.80	3.90	n.a.	n.a.
RT07 (only SAD)	1.20	2.10	3.30	n.a.	n.a.
RT06 (diarization)	4.40	2.30	6.70	4.10	10.81
RT07 (diarization)	4.50	1.50	6.00	2.50	8.51

Table 3. *Performance of the RT06 and RT07 speech/non-speech detectors on the RT07s Eval data. In the first two rows, only the SAD segmentation is scored. The last two rows show the results of the RT07 diarization system using either the RT06 speech/non-speech system or the RT07 speech/non-speech system.*

Smoothing SAD In previous years, we have tuned our speech/non-speech detectors by minimizing the SAD error on a development set. One of the steps that helps in minimizing the SAD error is 'smoothing' the output (NIST provides scripts to do this). During this process, short non-speech segments (shorter than 0.3s) are removed from the segmentation. Smoothing helps to reduce the SAD error because the reference segmentation is smoothed as well, and so these little fragments of non-speech will be regarded as missed speech if no smoothing is performed. On the other hand, adding these short non-speech segments to the speech data that is processed by the speaker diarization system will most likely increase the DER. The non-speech will be assigned to one or more clusters and will "muddy" the data pool, forcing the GMMs to be less specific for a particular speaker. Therefore, this year we decided to use the unsmoothed speech/non-speech segmentation as input to our diarization system and perform smoothing after the diarization process is finished. The improvement over using the smoothed speech/non-speech segmentations on the test set was marginal. On the conference room MDM task, using the smoothed segmentation resulted in a diarization error of 9.03%, and so the improvement by using the unsmoothed speech/non-speech input was only 0.52% absolute.

Blame assignment In order to find out what part of our system is contributing most to the total DER, we conducted a cheating experiment. Instead of using the automatically generated speech/non-speech segmentation, we used the reference segmentation as input for our diarization system. Table 4 contains the error rates

of our MDM and SDM submissions and the results of the cheating experiments. All results are scored with and without overlap.

	%Miss	%FA	%Spkr	%DER
MDM -ref +ovlp	**4.5**	**1.5**	**2.5**	**8.51**
MDM +ref +ovlp	3.7	0.0	3.8	7.47
MDM -ref -ovlp	0.9	1.6	2.6	5.11
MDM +ref -ovlp	0.0	0.0	3.9	3.94
SDM -ref +ovlp	**5.0**	**1.8**	**14.9**	**21.74**
SDM +ref +ovlp	3.7	0.0	12.8	16.51
SDM -ref -ovlp	1.4	2.0	14.7	18.03
SDM +ref -ovlp	0.0	0.0	12.7	12.75

Table 4. *DER for the MDM and SDM submissions. The rows in bold show the results of the actual submissions. They are scored with overlap and make use of our speech/non-speech segmentation. The systems marked with -ovlp/+ovlp are scored with/without overlap and the systems marked with -ref/+ref make use of the automatic speech/non-speech segmentation or of the reference speech/non-speech segmentation.*

Even if the reference segmentation is used, the percentage of missed speech will not be zero. This is because our diarization system is only able to assign a speech fragment to one single speaker and thus, when scoring with overlap speech, all overlapping speech will be missed. As can be seen in the second row of table 4 the error due to missed overlapping speech is 3.7%. The total error due to missed speech and false alarms is 6.0%. Subtracting the error due to overlap leaves the error contribution of our speech/non-speech system: 2.3%. The remaining 3.8% of the total DER is caused by the diarization step (speaker error). Note that the percentages change slightly if scored without overlap because ignoring segments with overlap will decrease the total amount of speech, which is part of the DER calculation.

The same blame assignment can be done for the SDM task. The error because of missed overlapping speech for the SDM task is 3.7%, and the error due to the speech/non-speech detector is 3.1% (3.4% if scored without overlap). The speaker error caused by the diarization system is 14.9%.

Noise Filtering In a series of experiments, we tested how much the system gains from applying Wiener filtering. Wiener filtering is normally applied to the audio used for the speech/non-speech detector, and on the audio that is used to create MFCC features, and on the audio that is used to calculate the delay features. Table 5 shows the results of several experiments where we omitted filtering for one or more of these components. It shows that filtering helps to reduce the DER considerably. Although it seems that filtering the audio for speech/non-speech helps the most, the SAD error on unfiltered audio only increases marginally (from 3.3% to 3.4%).

Where do we apply Wiener filtering?	%DER
Nowhere	15.80
Speech/non-speech	10.54
Speech/non-speech and MFCC	12.99
Speech/non-speech and Delays	13.70
All components	8.51

Table 5. *DER for the MDM submission (bottom row) and for the experiments where Wiener filtering is omitted in one or more of the components.*

Delay Features This year we used the algorithm introduced in [7] to automatically determine stream weights for the MFCC and delay feature streams. In last year's submission, the weights were fixed to 0.9 and 0.1. We have conducted an experiment on this year's evaluation data where the weights were fixed (as was done last year) in order to determine if the adaptive weighting was the right choice for the evaluation data. On the MDM conference meeting task the DER was 9.29% using the RT 2006 fixed weights. Thus, the new algorithm improved the DER by 0.78% absolute.

The gap between our results in the SDM task and MDM task is considerably large. This performance difference could be because it is not possible to use the second (delay) feature stream for SDM. To test this hypothesis we have ran the MDM data using only the MFCC stream. The diarization error of this experiment is 14.02%. A difference of 5.51% DER absolute.

4 Conclusions

In this paper, we have presented the ICSI RT07s speaker diarization system. This year, we introduced a new speech/non-speech detector that is able to filter out audible non-speech without the need for models trained on "outside" data. This new speech/non-speech system reduced false alarm errors by 0.7% absolute compared to our RT06s speech/non-speech system. Post evaluation experiments showed that by reducing the false alarms, the diarization system also performed better (2.3% DER absolute).

Other post evaluation experiments showed that the use of cross-channel delays as a second feature stream (for the MDM task) improved the system considerably resulting in a gain of 5.51% DER absolute. We also observed that omitting noise filtering in either one of the feature streams decreases the performance of the system by up to 7.29% absolute. We obtained modest improvements (0.52% DER absolute) in system performance by using unsmoothed speech/non-speech segmentations as input to the diarization system. We also achieved another modest improvement (0.78% DER absolute) by dynamically tuning the stream weights as proposed by [7] rather than using fixed stream weights.

The gap in performance of our system between the SDM and MDM tasks is striking. Our post evaluation experiments showed that the errors due to missed

overlapping speech and misclassified speech/non-speech are comparable for the two tasks. Thus, the main difference in performance is caused by the diarization system itself (3.8% DER for MDM and 14.9% DER for SDM). We believe that our SDM system can be improved considerably by introducing additional feature streams, similar to what we used in the MDM system. Of course these additional streams would not be based on delays since there is only a single microphone in the SDM condition, but we believe that we could use other acoustic features (e.g. PLP or RASTA features), or even the output of other speaker diarization systems as additional feature streams. For the next evaluation we will concentrate on finding suitable features to add to the primary MFCC feature stream.

Finally, because of the "flakiness" of the diarization error rate, this year we performed all of our development work using a much larger set of recordings (21 in total) than we have used in past evaluations. We believe that using this larger set of data helps to reduce some of the flakiness, thus leading to better decisions about system design and tuning.

5 Acknowledgments

This work was partly supported by the European Union 6th FWP IST Integrated Project AMIDA (Augmented Multi-party Interaction with Distant Access, FP6-506811), by the Swiss National Science Foundation through NCCR's IM2 project, and by the MultimediaN project (http://www.multimedian.nl). MultimediaN is sponsored by the Dutch government under contract BSIK 03031.

The work was also partly funded by DARPA under contract No. HR0011-06-C-0023 (approved for public release, distribution is unlimited).

References

1. Reynolds, D., Torres-Carrasquillo, P.: Approaches and applications of audio diarization, Philadelphia, PA (2005) 953–956
2. Shaobing Chen, S., Gopalakrishnan, P.: Speaker, environment and channel change detection and clustering via the bayesian information criterion. In: Proceedings DARPA Broadcast News Transcription and Understanding Workshop, Virginia, USA (1998)
3. Wiener, Norbert: Extrapolation, Interpolation, and Smoothing of Stationary Time Series. Wiley (1949)
4. Adami, A., Burget, L., Dupont, S., Garudadri, H., Grezl, F., Hermansky, H., Jain, P., Kajarekar, S., Morgan, N., Sivadas, S.: Qualcomm-icsi-ogi features for asr. (2002)
5. Huijbregts, M., Ordelman, R., de Jong, F.: Speech-based annotation of heterogeneous multimedia content using automatic speech recognition. Technical Report TR-CTIT-07-30, Enschede (2007) publisher=Centre for Telematics and Information Technology, University of Twente, number of pages=11.
6. Huijbregts, M., Wooters, C., Ordelman, R.: Filtering the unknown: Speech activity detection in heterogeneous video collections. In: Interspeech, Antwerp, Belgium (2007)

7. Anguera, X.: Robust Speaker Diarization for Meetings. PhD thesis, Universitat Politecnica De Catalunya (2006)
8. Wooters, C., Fung, J., Peskin, B., Anguera, X.: Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system. In: Fall 2004 Rich Transcription Workshop (RT04), Palisades, NY (2004)
9. Ajmera, J., Wooters, C.: A robust speaker clustering algorithm, US Virgin Islands, USA (2003)
10. Ajmera, J., McCowan, I., Bourlard, H.: Robust speaker change detection. IEEE Signal Processing Letters **11**(8) (2004) 649–651
11. NIST: Rich Transcription Evaluations, website: http://www.nist.gov/speech/tests/rt (2007)
12. Mirghafori, N., Wooters, C.: Nuts and flakes: A study of data characteristics in speaker diarization, Toulouse, France (2006)