Exploring methods of improving speaker accuracy for speaker diarization

Mary Tai Knox[1,2], *Nikki Mirghafori*[1], *Gerald Friedland*[1]

[1]International Computer Science Institute, Berkeley, California, USA
[2]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA
{knoxm, nikki, fractor}@icsi.berkeley.edu

Abstract

The focus of this work is to improve the speaker diarization error rate, and more specifically the speaker error rate. We investigate two methods of improving the speaker error rate: modifying the minimum duration constraint and incorporating novel purification techniques. First, in the final step of the speaker diarization algorithm we replace the minimum duration constraint with a simple smoothing algorithm, which averages the log-likelihoods for each of the hypothesized speakers. This method improves the speaker error rate by 12% relative for the MDM condition. Second, we utilize the difference between the largest and second largest log-likelihoods to identify frames which are believed to be correct (or "pure"). The difference value is shown be more effective at separating correct frames from incorrect frames than the previously used maximum log-likelihood value. Using only the "pure" frames, the cluster models are retrained and segmentation is performed using the above mentioned smoothing technique. The proposed purification and smoothing reduces the speaker error rate over the baseline; however, it is worse than performing the smoothing step alone.

Index Terms: speaker diarization, cluster purification, temporal smoothing

1. Introduction

The goal of speaker diarization is to partition an audio signal into speaker homogeneous speech regions, as shown in Figure 1, where the number of speakers as well as the speaker identities are not known a priori. Speaker diarization has many applications, including speaker adaption for automatic speech recognition [1], audio indexing [2, 3], and speaker localization [4].

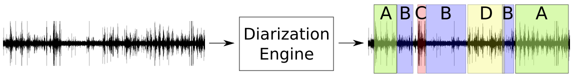

Figure 1: Overview of speaker diarization. From an input audio signal, segment the signal into nonspeech and speech segments, the latter labeled by speaker (e.g., A, B, C, D).

In this work, we improve upon previous speaker diarization systems, focusing improvement on reducing the speaker error rate. The speaker error rate, a component of the Diarization Error Rate (DER), is the percent of speech time that the hypothesized speaker does not match the reference speaker. Here, two approaches are investigated to reduce the speaker error rate: improving upon the minimum duration constraint and utilizing cluster purification.

In our previous work [5], it was shown that a significant amount of errors occurred during short segments as well as near speaker change points. It was hypothesized that maybe this is a result of the minimum duration constraint which does not allow speaker changes to occur within t_{mindur} seconds of speech. In this work, we investigate an alternative to the minimum duration constraint, namely median and mean smoothing over the log-likelihoods for each hypothesized speaker. While the minimum duration constraint is useful for eliminating rapid speaker changes, it puts a sharp threshold on the duration between speaker change points (often 1.5 to 2.5 seconds of speech [6]). Utilizing a smoothing approach lessens this restriction while still reducing the ability to have rapid speaker changes.

We also investigate cluster purification methods, where cluster models are trained only on the "pure" data, to improve speaker error rates. Cluster purification methods have shown to improve diarization results [7, 8]. In [7], models are first trained according to uniform initialization. Then the data in each cluster is split into 0.5 second segments. The top 25% of the segments in each cluster are labeled and the models for each cluster are retrained. More segments are iteratively labeled and the models are retrained until all of the data is labeled and included in the models. Another method of purification is used in [8], where the authors use the top 55% of segments to retrain speaker models. The latter work utilizes the purification method at the end of the algorithm while the former algorithm performs "purification" in the initial step. Note that the system described in [8] is a top-down speaker diarization system while the system in [7] is a bottom-up system.

In this work, a novel method is utilized to determine which data to use to retrain the models. As opposed to previous work which uses the data that best fits the Gaussian Mixture Model (GMM) (the data associated with the highest log-likelihoods for each cluster), in this work the models are retrained on the data with the highest difference in log-likelihoods for the best matched cluster and the second best matched cluster. In other words, it uses data which better matches one cluster over all other clusters.

This paper is outlined as follows: in Section 2 we describe the relevant background information, in Section 3 we discuss and analyze the preliminary findings on the development set, in Section 4 we discuss the evaluation set results, and in Section 5 we give our conclusions as well as areas of future work.

2. Background

2.1. Baseline diarization system

There are a number of methods used to perform speaker diarization [7, 8, 6, 9, 10, 11, 12] . The baseline system in this work is the ICSI speaker diarization system used in the NIST Rich Transcription 2009 (RT-09) evaluation. A more in depth description of the system is given in [6]. A short description is given below.

The ICSI speaker diarization system uses a Hidden Markov

Model (HMM) - Gaussian Mixture Model (GMM) agglomerative hierarchical clustering approach [13, 14]. This is a bottom-up approach where clusters are iteratively merged until each cluster represents a hypothesized speaker in the meeting. The algorithm is based on an HMM where each state (or cluster) is modeled as a GMM.

More specifically, speech detection is performed first and the speech regions are initially assigned to k clusters. A long-term feature based initialization is used to determine the value of k and the initial segmentation [14].

After the initial segmentation, GMM parameters are trained and the input stream is re-segmented using the Viterbi form of the Expectation Maximization (EM) algorithm. Note that for segmentation, a minimum duration constraint of 2.5 seconds of speech is used to prevent rampant speaker changes [13]. More specifically, each state has a number of substates, which span t_{mindur} seconds and have the same probability density function.

After updating the models of each of the clusters, the next step is to determine which two clusters to merge. This is done using the delta Bayesian Information Criterion (ΔBIC) [15], which is computed for each pair of clusters.

Once two clusters are merged, the GMM parameters are retrained and Viterbi decoding is performed to output the most probable segmentation (with a 2.5 second minimum duration constraint). The merging, retraining, and re-segmentation is repeated iteratively until the stopping criterion is met. After which, a final re-segmentation/re-training step is performed where the minimum duration is reduced to 1.5 seconds. As a final smoothing step, if the speakers immediately preceding and following a short non-speech segment (less than 0.5 seconds) are the same, the non-speech segment is relabeled as a speech segment spoken by the same speaker. This step is later referred to as *gapsmoothing*.

2.2. Scoring metrics

There are two main metrics used in this work: Diarization Error Rate (DER) and speaker accuracy. The DER is the sum of the per speaker false alarm time (overestimating the number of speakers), miss time (underestimating the number of speakers), and speaker error time (the hypothesized speaker(s) is (are) not matched to the appropriate speaker(s) in the reference) divided by the total speech time in an audio file, as shown in Equation (1). Note that if three people are speaking at the same time for a duration of one second, this results in three seconds of speech time. As done in the NIST evaluations, we scored the DER using a *no-score collar* of ±0.25 seconds [16] around reference segment boundaries. Since the focus in this study is on improving the speaker error rate, often times we simply report the speaker error time T_{SPKR}.

$$\text{DER} = \frac{T_{FA} + T_{MISS} + T_{SPKR}}{T_{SPEECH}} \tag{1}$$

Speaker accuracy is also utilized to evaluate performance in this study. It is simply the amount of time the hypothesized speaker is correctly labeled divided by the total time at least one speaker is speaking. The speaker diarization system used in this study does not address the overlapped speaker problem; and therefore, assigns at most one speaker to any time instance. In the case that more than one person is speaking at the same time, it is correct (in terms of speaker accuracy) if the hypothesized speaker corresponds to one of the people speaking. The denominator for speaker accuracy differs from that used to compute the DER since overlapped speech is only counted once.

2.3. Data

This work is performed on the NIST Rich Transcription (RT) dataset. More specifically, we use only meeting domain recordings. Recordings from the meeting domain have been the focus of the latest RT evaluations and contain spontaneous speech, which is representative of "real-world" interactions and challenging due to disfluencies.

The data was split into two partitions: a development set and a test set. The development set consists of 28 meeting recordings from RT evaluations prior to RT-09. The test set consists of 7 meeting recordings from the latest evaluation set, RT-09. Both the multiple distant microphone (MDM) and single distant microphone (SDM) conditions are investigated.

3. Experimental analysis

In this section, we describe the experiments we performed on the development set. The results of this experimental analysis are used to determine the parameters of the final system.

3.1. Minimum duration versus log-likelihood smoothing

We compare the speaker error time (T_{SPKR}) for four different approaches to reducing rapid speaker changes. The first approach is to modify the minimum duration constraint used within the algorithm (the original system uses 2.5 seconds) while keeping the last iteration at the default 1.5 seconds. The second approach changes the minimum duration constraint used in the last iteration of the algorithm (the original system uses 1.5 seconds) while using the default 2.5 second minimum duration constraint within the algorithm. The third and fourth approaches apply mean and median smoothing (over a varied number of frames) to the final log-likelihoods for each hypothesized speaker. The new segmentation is performed on a per frame basis, where the hypothesized speaker has the highest mean or median smoothed log-likelihood. The experiments are performed for both the MDM and SDM conditions and the results are shown in Tables 1 and 2, respectively.

Table 1: MDM – Speaker error time (in seconds). Values denoted with * have combined miss and false alarm rates greater than 6.0% (due to gapsmoothing).

Min dur or smooth time	Min dur	Min dur last iter	Mean smoothing	Median smoothing
0.0		4598*	1083	1083
0.5	1611	544*	480	529
1.0	914	505	**429**	446
1.5	880	527	456	447
2.0	960	589	503	478
2.5	527	666	568	519
3.0	727	730	625	566
3.5	1102	817*	691	623

As shown in Tables 1 and 2, determining the speaker via mean smoothing the log-likelihood scores is the best and second best method for the MDM and SDM conditions, respectively. This method results in a 18.5% and 3.2% relative decrease in speaker error rate over the baselines (526.79 seconds and 1985.49 seconds) for the MDM and SDM conditions, respectively. The DER decreases from 9.6% to 9.0% for the MDM condition and from 19.6% to 19.3% for the SDM condition. Though the results for the SDM condition are not as dramatic as the MDM condition, the smoothing results are consis-

Table 2: SDM – Speaker error time (in seconds). Values denoted with * have combined miss and false alarm rates greater than 6.4%.

Min dur or smooth time	Min dur	Min dur last iter	Mean smoothing	Median smoothing
0.0		5355*	6303	6303
0.5	7457	2312*	2288	2671
1.0	3949	2069	1945	2089
1.5	2839	1985	**1923**	1991
2.0	2218	2017	1955	2002
2.5	1985	2133	1994	2025
3.0	1939	2109	2039	2043
3.5	**1875**	2118	2095	2081

tently better for shorter smoothing values. We hypothesize that the SDM results do not improve as dramatically as the MDM results because the speaker models are not as good. As an aside, the values annotated with an asterisk contain a higher combined miss and false alarm error rate than the other values in the table. This is due to gapsmoothing, which is described in Section 2.1.

3.2. Identifying "pure" frames

In this part of the analysis, we first evaluate the effectiveness of separating the correct frames from the incorrect frames for five attributes based on the log-likelihood scores: maximum, variance, unnormalized entropy, posterior probability, and difference between the largest and second largest log-likelihood scores. The log-likelihoods for each of the final clusters are computed and mean smoothed over a number of durations. Then the five attributes summarized below are computed for each frame.

- **Maximum:** The maximum smoothed log-likelihood score.
- **Variance:** The variance of the smoothed log-likelihood scores for all of the final clusters.
- **Entropy:** An unnormalized entropy of the smoothed log-likelihood scores for all of the final clusters. More specifically, let $p(x_t|\theta_k)$ be the probability of the feature vector x at time t given θ_k (the GMM parameters of cluster k). Then the unnormalized entropy of the log-likelihoods is defined as,

$$\hat{H}(p(x_t|\theta)) = -\sum_{k=1}^{n} p(x_t|\theta_k) \log p(x_t|\theta_k), \quad (2)$$

  where n is the number of final clusters.
- **Posterior Probability:** The maximum posterior probability $p(\theta_k|x_t)$. Note that $p(\theta_k) = 1/n$ since in our setup, we assume each cluster is equally likely.

$$\max_k p(\theta_k|x_t) = \max_k p(x_t|\theta_k) / \sum_{k=1}^{n} p(x_t|\theta_k) \quad (3)$$

- **Difference:** The difference between the largest and second largest smoothed log-likelihood scores.

In order to measure the strength of each attribute, we utilize the Receiver Operating Characteristic (ROC) curve. More specifically, we compute the Area Under the Curve (AUC) value for each of the five log-likelihood attributes. Tables 3 and 4 show the ROC AUC values for the various log-likelihood attributes. The attributes which better separate the correct and incorrect classes have larger ROC AUC values. Note that the correct and incorrect labels are based on the baseline system results. We will incorporate making a frame level decision based on the smoothed log-likelihood score in the next set of experiments.

Table 3: MDM – Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) values for the various mean smoothed log-likelihood attributes.

Smooth time (s)	Diff	Post	Var	Max	Entr
0.0	0.76	0.76	0.65	0.63	0.62
0.5	0.83	0.83	0.68	0.67	0.66
1.0	0.84	0.84	0.70	0.68	0.67
1.5	0.84	0.85	0.70	0.69	0.67
2.0	0.84	0.84	0.71	0.69	0.67
2.5	0.83	0.83	0.71	0.68	0.67
3.0	0.82	0.82	0.71	0.68	0.66
3.5	0.81	0.81	0.71	0.67	0.66

Table 4: SDM – Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) values for the various mean smoothed log-likelihood attributes.

Smooth time (s)	Diff	Post	Var	Max	Entr
0.0	0.57	0.57	0.50	0.53	0.52
0.5	0.73	0.72	0.53	0.60	0.56
1.0	0.77	0.75	0.54	0.62	0.57
1.5	0.79	0.76	0.55	0.63	0.57
2.0	0.79	0.76	0.55	0.63	0.57
2.5	0.79	0.76	0.55	0.63	0.57
3.0	0.78	0.76	0.56	0.63	0.57
3.5	0.78	0.76	0.56	0.63	0.57

From Tables 3 and 4, we observe that for both the MDM and SDM conditions the difference and posterior probability have the largest ROC AUC values, and therefore perform the best. For the SDM condition, the difference performs better than the posterior probability and for the MDM condition the two attributes result in nearly the same ROC AUC values. Since the performance is so similar for the difference and the posterior probability, with the difference performing slightly better than the posterior probability for the SDM condition and essentially the same for the MDM condition, we will investigate the difference attribute in further detail. The maximum log-likelihood attribute is the third best performing attribute for the SDM condition and fourth best for the MDM condition. For the MDM condition, the variance log-likelihood attribute is third best in terms of the ROC AUC; however, for the SDM condition, the variance is worst performing attribute. Therefore, it will not be examined further. We also investigated the mean of the smoothed log-likelihood scores for all of the clusters and found that it performed poorly.

We further analyze the difference and maximum attributes to determine their strength in separating correct frames from incorrect frames, which is useful for performing cluster purification. The difference attribute performed the best and since previous work [7, 8] relies on using the maximum log-likelihood scores to determine which frames should be used for cluster purification, we compare the results when using the maximum log-likelihood to the difference between the largest and second largest log-likelihood. We compute the speaker accuracy, where the hypothesized speaker is the speaker with the largest smoothed log-likelihood score, when scoring the frames which had the highest per cluster difference or maximum scores. Figures 2 and 3 show the results for all scored time (i.e. ignoring "collar" time) for the MDM and SDM conditions. More specifically, the figures show that for both the MDM and SDM conditions, the speaker accuracy for the best difference scores is better than the speaker accuracy for the best maximum log-likelihood scores (particularly for the very best scores for each of the two attributes).

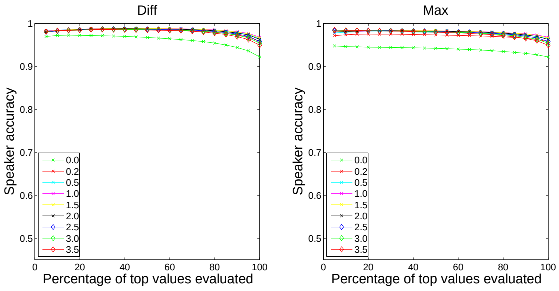

Figure 2: MDM – Speaker accuracy for the per cluster top x% of difference between the largest and second largest mean smoothed log-likelihood values (on the left) and maximum log-likelihood values (on the right) shown for a variety of smoothing durations as denoted in the legend.

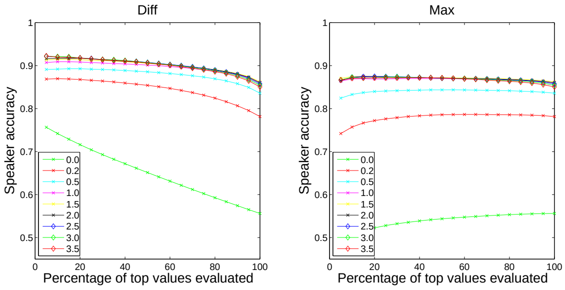

Figure 3: SDM – Speaker accuracy for the per cluster top x% of difference between the largest and second largest mean smoothed log-likelihood values (on the left) and maximum log-likelihood values (on the right) shown for a variety of smoothing durations as denoted in the legend.

3.3. Cluster purification

Based on the previous results, we investigate the use of the difference between the largest and second largest log-likelihood values to determine which frames to use to retrain the cluster models. Log-likelihood values are smoothed over 1.0 seconds and 1.5 seconds for MDM and SDM, respectively. The smoothing values were determined according to the results found in Tables 3 and 4, namely we chose the smallest duration for which the ROC AUC values are large. Figure 4 shows the final speaker error results when using a variable amount of data (according to the top difference scores) to retrain the speaker models. For comparison, we have also shown the results when using the maximum log-likelihood to determine which frames to train the "purified" models on. Similar to previous purification work [7, 8], each cluster is split into 0.5 second segments. In this work, the scores are averaged for 0.5 second non-overlapping windows. Also, for the MDM condition only the Mel-Frequency Cepstral Coefficients (MFCCs) are "purified". The GMMs trained on delay features are kept the same. This is because there already is not much diversity in the delay feature values. Previous work also does not purify GMMs trained on delay features. Based on the results shown in Figure 4 we see that retraining on the best per cluster difference values results in a lower amount of speaker error rate than retraining on the best per cluster maximum log-likelihood scores. However, decreasing the amount of training data used in the final models and then mean filtering does not perform as well as mean filtering alone.

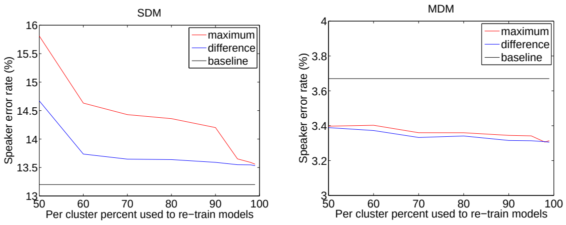

Figure 4: Speaker accuracy after retraining models on the top x% of difference between the largest and second largest mean smoothed log-likelihood values (blue) and maximum log-likelihood values (red) for the MDM (left) and SDM (right) conditions.

4. Test Set Results

Since cluster purification on the last iteration was not found to improve results, we simply perform mean filtering at the last iteration. For the MDM condition, the log-likelihoods are mean filtered over a 1.0 second window and for SDM this window is increased to 1.5 seconds. For MDM, the amount of speaker error is reduced from 430.7 seconds to 379.4 seconds, which is an 11.9% relative improvement. This results in a DER of 17.3% and 16.5%, respectively. For the SDM condition, the result is not as dramatic. The speaker error time is reduced from 1086.5 seconds to 1055.7 seconds, or a 2.8% relative improvement, and the DER decreased from 29.2% to 28.6%.

5. Conclusions and Future Work

In conclusion, we investigate two methods of reducing the speaker error rate for our speaker diarization system. The first method involves averaging the log-likelihood scores for each hypothesized speaker. This was performed on the last iteration of the speaker diarization algorithm and results in an 11.9% relative improvement for the MDM condition and a 3% improvement for the SDM condition. We also investigate the usefulness of the difference between the largest and second largest log-likelihood in separating correct and incorrect frames. We found that the difference attribute performed better than the maximum log-likelihood in terms of identifying correct frames. However, for our diarization algorithm cluster purification on the last iteration did not reduce the DER.

For future work, we would like to investigate results when log-likelihood averaging is incorporated throughout the speaker diarization algorithm (instead of only on the final iteration). Also, other researchers have found purification based on the maximum log-likelihoods to improve results [7, 8]. Since we have shown that the difference feature better separates the correct and incorrect classes, perhaps it would perform better for purification on those systems as well.

6. References

[1] A. Stolcke, G. Friedland, and D. Imseng, “Leveraging speaker diarization for meeting recognition from distant microphones,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, March 2010, pp. 4390–4396.

[2] G. Friedland, L. Gottlieb, and A. Janin, “Joke-o-mat: browsing sitcoms punchline by punchine,” in *ACM Multimedia Conference*, New York, New York, 2009, pp. 1115–1116.

[3] R. Mertens, P.-S. Huang, L. Gottlieb, G. Friedland, and A. Divakaran, “On the applicability of speaker diarization to audio concept detection for multimedia retrieval,” in *IEEE International Symposium on Multimedia (ISM)*, December 2011, pp. 446–451.

[4] G. Friedland, C. Yeo, and H. Hung, “Dialocalization: Acoustic speaker diarization and visual localization as joint optimization problem,” *ACM Transactions on Multimedia Computing, Communications and Applications, Special Issue on Sensor Fusion*, vol. 6, no. 4, November 2010.

[5] M. Knox, N. Mirghafori, and G. Friedland, “Where did i go wrong?: Identifying troublesome segments for speaker diarization systems,” in *Interspeech*, Portland, Oregon, 2012.

[6] G. Friedland, A. Janin, D. Imseng, X. Anguera Miro, L. Gottlieb, M. Huijbregts, M. Knox, and O. Vinyals, “The ICSI RT-09 speaker diarization system,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 2, pp. 371–381, February 2012.

[7] T. Nguyen, H. Sun, S. Zhao, S. Khine, H. Tran, T. Ma, B. Ma, E. Chng, and H. Li, “The IIR-NTU speaker diarization systems for RT 2009,” in *RT’09, NIST Rich Transcription Workshop*, Melbourne, Florida, 2009.

[8] S. Bozonnet, N. Evans, and C. Fredouille, “The LIA-Eurecom RT’09 speaker diarization system: Enhancements in speaker modelling and cluster purification,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, March 2010.

[9] M. Huijbregts, “Segmentation, diarization and speech transcription: Surprise data unraveled,” Ph.D. dissertation, University of Twente, Enschede, Netherlands, 2008.

[10] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, “A sticky HDP-HMM with application to speaker diarization,” *Anals of Applied Statistics*, June 2011.

[11] D. Vijayasenan, F. Valente, and H. Bourlard, “Multistream speaker diarization of meeting recordings beyond MFCC and TDOA features,” *Speech Communication*, vol. 54, pp. 55–67, January 2012.

[12] S. Shum, N. Dehak, and J. Glass, “On the use of spectral and iterative methods for speaker diarization,” in *Interspeech*, Portland, Oregon, September 2012.

[13] C. Wooters and M. Huijbregts, “The ICSI RT07s speaker diarization system,” in *Proceedings of the RT07 Meeting Recognition Evaluation Workshop*, 2007.

[14] D. Imseng and G. Friedland, “Tuning-robust initialization methods for speaker diarization,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, pp. 2028–2037, 2010.

[15] J. Ajmera and C. Wooters, “A robust speaker clustering algorithm,” in *Proceedings of IEEE Speech Recognition and Understanding Workshop*, St. Thomas, US Virgin Islands, 2003.

[16] J. Ajot and J. Fiscus, “RT-09 speaker diarization results,” Melbourne, Florida, May 2009.