THE LINCOLN SPEAKER RECOGNITION SYSTEM: NIST EVAL2000

D.A. Reynolds, R.B. Dunn and J.J. McLaughlin
MIT Lincoln Laboratory, Lexington, MA, USA
{dar, rbd, jackm}@stll.mit.edu

ABSTRACT
This paper presents an overview of the Lincoln Laboratory systems fielded for the 2000 NIST speaker recognition evaluation (SRE00). In addition to the standard one-speaker detection tasks, this year's evaluation, as in 1999, included multi-speaker spoiled data with detection, tracking, and segmentation. The design approach for the Lincoln system in SRE00 was to develop a set of core one-speaker detection and multi-speaker clustering tools that could be applied to all the tasks. This paper will describe these core systems, how they are applied to the SRE00 tasks and the results they produce. Additionally, a new channel normalization technique known as handset-dependent test-score norm (HThnorm) is introduced.

1 INTRODUCTION
In this paper we describe the Lincoln systems for the 2000 NIST speaker recognition evaluation (SRE00). This evaluation was designed to focus on the tasks of speaker detection, segmentation, and tracking. These tasks are posed in the context of conversational telephone speech and for limited training data. In section 2 the systems fielded by Lincoln are described. Following the evaluation, there were two improvements made to the systems. The first modification, an improvement to the core one-speaker detection system, is a new channel normalization technique known as handset-dependent test-score norm (HThnorm). The second improvement, for the two-speaker detection and tracking systems was to combine internal and external segmentation systems to improve performance. The improvements are described in section 3 and system performance is then presented in section 4.

1.1 Tasks
The four primary tasks comprising the NIST SRE00 are defined as follows:

1. One-Speaker Detection: determine if a specified target speaker is speaking during a sample of speech which contains one speaker.

2. Two-Speaker Detection: determine if a specified target speaker is speaking during a sample of speech which contains two speakers.

3. Two-Speaker Tracking: determine when, if at all, a specified target speaker is speaking during a sample of speech containing two speakers.

4. Speaker Segmentation: determine the time intervals during which each participant in a conversation is speaking (the number of participants may, or may not, be known).

1.2 Data
The primary data for past evaluations, SRE96 through SRE99, have been the Switchboard-1 and Switchboard-2 Corpora [2]. These corpora are made up of unscripted telephone conversations between native English speakers with two people per conversation. The evaluation data for SRE00 was selected from the SRE97 and SRE98 data and as such these data were not to be used for system development. The data from SRE96 and SRE99 were available for development and we chose to use SRE06 data to train a universal background model and compute normalization parameters. During development we used SRE99 data to evaluate our system performance. The N-speaker segmentation task required an additional database, CALHOME [3], which contains unscripted telephone conversations in six different languages with two to seven people participating in each conversation, was used for this task.

2 THE LINCOLN SYSTEM
The Lincoln system is made up of a core single-speaker detector and additional multi-speaker processing tools. The single-speaker detector is described in section 2.1 and then the multi-speaker processing tools and the system configurations for multi-speaker tasks are described in 2.2.

2.1 GMM-UBM one-speaker detector
The basic single-speaker detector is a likelihood ratio detector with target and alternative probability distributions modeled by Gaussian mixture models (GMMs) as shown in Figure 1. As introduced by Lincoln Laboratory in 1996 and used successfully in previous evaluations, a Universal background model (UBM) GMM is used as the alternative hypothesis model, and from this target models are derived using Bayesian adaptation (also known as Maximum A-Posteriori (MAP) training) [3]. The scores are normalized such that a single-speaker-independent threshold can be used for detection.

![Figure 1: GMM-UBM likelihood ratio detector.](image-url)

The front-end processing for the system is as follows. A 19-dimensional mel-cepstral vector is extracted from the speech signal every 10 ms using a 20 ms window. The mel-cepstral vector is computed using an unweighted triangular filterbank on the DFT spectrum. Bandlimiting is then performed by only retaining the filterbank outputs from the frequency range 300–3158 Hz. Cepstral vectors are processed with RASTA filtering to mitigate linear channel bias effects. Delta cepstra are then computed over a ±2 frame span and...
appended to the cepstra vector producing a 38-dimensional feature vector. Lastly, the feature vector stream is processed through an adaptive, energy-based speech detector to discard low-energy vectors.

The UBM is a 2018 mixture gender-independent, handset-independent GMM trained using 4.6 hours of data selected from SRE96 to be approximately evenly divided between sex and handset type. On development test data (SR99) slightly better performance was obtained when using a gender-dependent UBM matched to the gender of the target speaker. However, it was found that this gender-matched UBM performed significantly worse under cross-gender test conditions. While cross-gender testing is not part of the single-speaker evaluation it does occur in the two-speaker evaluations. In order to use a common system on one- and two-speaker tasks a gender-independent UBM was used.

Target models are derived by Bayesian adaptation (also MAP estimation) of the UBM parameters using the designated two minutes of training data. Only the mean vectors are adapted as this has been observed to provide better performance. The amount of adaptation of each mixture mean is data dependent. Details of the adapted GMM-UBM system can be found in [5].

The likelihood ratio scores in Figure 1 are normalized first with Thurm [4] and then Hnorm [3]. Thurm is a new technique introduced in 1990 by Ensign where scores from a collection of fixed non-target models are used to normalize a target model score for a test file. The target model score normalization is accomplished by subtracting the mean and dividing by the standard deviation of the non-target model scores per test file. This contrasts with Hnorm where scores from a handset-dependent collection of fixed non-target speech samples are used to normalize a speaker model. In our system scores are normalized first with Thurm and then with Hnorm as it was found that using Hnorm on Thurm scored provided better performance than computing Thurm scores from Hnorm speaker models.

2.2 Multi-speaker processing

The multi-speaker processing tools consist of an initial segmentation, a clustering algorithm, and an iterative resegmentation algorithm (Figure 2). The multi-speaker speech is partitioned into small segments, each roughly 1 second in duration and presumed to contain only one speaker. These segments are accumulated into clusters, using a hierarchical agglomerative clustering technique, where the resulting clusters are also presumed to contain only one speaker. Multi-speaker detection is accomplished by applying the one-speaker detection system to each cluster. The tracking and segmentation tasks use an additional iterative resegmentation process which improves segment boundaries.

2.2.1 Multi-Speaker Processing Tools

The multi-speaker processing tools differ in the end processing than the standard speaker detection system in order to take advantage of channel differences between the speakers in the signal. As in one-speaker detection, a cepstral vector is extracted from the speech signal every 10 ms using a 20 ms window, however, bandlimiting is not used so the vector dimension is increased to 23. Channel equalization is not applied nor are delta cepstral features used.

The first task after front end processing is to segment the continuous stream of feature vectors from the file into segments which each contain at most one speaker. Development experiments were performed using a Bayesian information criteria (BIC) [6] based speaker change detection system for segmentation; however, it was found that equivalent performance was obtained from a simpler segmentation. In this second method, an adaptive, energy-based speech detector with settings tuned to detect only long pauses was used to create intervals which were further segmented into fixed length 100 frame (1 sec) segments prior to clustering.

The segments are then grouped into clusters using a hierarchical agglomerative clustering technique based on work published in [7]. To initialize the algorithm each segment is placed in its own cluster. Agglomerative clustering then proceeds by computing the pairwise distance between all clusters and merging the two clusters with the minimum distance. This is repeated until the desired number of clusters is obtained or a stopping criteria is met. The pairwise distance between clusters is based on the ratio of two likelihoods: the likelihood that the two clusters are generated by two different speakers and the likelihood that the segments in the two clusters are generated by the same speaker. These likelihoods are computed using tied-GMM density functions. For each segment, mixture weights to a common, fixed set of Gaussians are estimated. We use a set of 64 Gaussians trained on the file being segmented. The use of tied-GMMs provides better density modeling for the segments than the standard approach of using a unimodal Gaussian density. When two clusters are merged, new mixture weights using the union of segments in both clusters are estimated and distances to the remaining clusters are recomputed. The final number of clusters is dependent on the particular task. When the number of speakers is known it may be set to an a priori value and when the number of speakers is not known it is automatically generated using a BIC-based criteria.

After clustering, a second pass, resegmentation of the file is performed as follows. The speech from each cluster is first used to train individual GMMs (adapted from the UBM used in the one-speaker detection system), the entire file is re-scored with cluster GMMs and per-frame likelihood scores are computed. These likelihood score sequences are smoothed using a 23 second window and the file is resegmented by associating each frame with the maximum scoring cluster-model. The cluster-models can then be reestimated with the resegmented data and the process iterated. In practice only a single pass was performed for this system since development results indicated a decrease in performance with multiple iterations.

2.2.2 Multi-Speaker Systems

The two-speaker detection system the clusters generated by the agglomerative clustering algorithm are processed by the one-speaker detection system (Figure 2). The maximum of the individual clusters' likelihood ratio scores is used as the likelihood score for the input file since determining the target speaker's presence in any one of the clusters is sufficient to determine his presence in the file. For the two-speaker detection standard clustering algorithm is stopped when the three clusters remain. This is thought to improve performance when one speaker predominates the conversation. Also, the same clusters are used for two-speaker track-
The noise and development results indicate that performance on this task would be reduced if only two clusters were used.

In the two-speaker tracking system the three clusters are resegmented using a single iteration of the resegmentation tool and the likelihood ratio scores from the two-speaker detection task are assigned to the corresponding resegmented clusters. Although it is possible to use the resegmented clusters to produce new likelihood scores, this does not improve performance so the likelihood scores computed from the original set of clusters (before resegmenting) are used.

There were two segmentation subtasks and the stopping criteria of the clustering algorithm was individually tailored to each task. In one task the number of speakers was known to be two so the clustering algorithm was set to stop when two clusters had been generated. In the second clustering task the number of speakers was unknown. For this task a BIC-based stopping criteria was used to determine when the clustering algorithm should stop. Using the development data (SRE09) we determined that the resegmentation would substantially reduce the segmentation error. However, only a single iteration was used as further iterations did not improve performance.

3 POST EVALUATION IMPROVEMENTS

Two significant improvements were made to the Lincoln system after the evaluation. The first improvement was to introduce handset-dependent test-score normalization (HT-norm). This is similar to Thurman except that for HTThom the speaker models used for normalization are selected to match the handset type as well as the gender of the target speaker.

The second significant improvement is for the two-speaker detection and tracking systems. The two basic types of systems introduced in [8], internal and external segmentation systems, are combined to improve overall performance. The detection and tracking systems submitted by Lincoln in SRE09 are characterized as internal segmentation systems because the frame-by-frame likelihood ratio scores internal to the CMU-DEM system were used to both segment the speech and then validate those segments with respect to target speaker models. The system described in this paper is characterized as an external segmentation system because the speech is partitioned into speaker homogeneous regions by a clustering algorithm which is external to the one-speaker detection system. The scores of the internal and external segmentation systems are combined with a simple weighted sum to improve performance of both the two-speaker detection and tracking systems.

4 RESULTS

The performance for the detection and tracking tasks is demonstrated by Detection Error Tradeoff (DET) curves. These curves show the relationship between misses and false alarms as the detection threshold is varied. The performance of segmentation systems is evaluated using a segmentation error metric provided by NIST. The metric used can be roughly described as one minus the ratio of the amount of correctly segmented speech to the total amount of speech (in terms of duration). In the segmentation task the speaker for each segment is not identified, therefore, when computing the segmentation error the optimal mapping of speakers to segments is used.

The performance of the one-speaker detection system in SRE09 is shown by the solid lines in Figure 3. The Equal-Error Rate (EER) measured over all the test data is 10.1%. In the primary pooling where all speaker models were trained on electret speech segments and all test segments were electret, the EER is reduced to 8.0%. The dashed lines in the figure show how the performance improves when HTThom is used along with a background model trained on SRE09 data and normalization parameters from SRE09 data. Most of the improvement is gained by the use of SRE09 data for normalization parameters. The dashed lines represent system performance that exceeds that of all systems submitted in SRE09.

The performance of the two-speaker detection system, shown in Figure 4, was the best of any in the evaluation. The performance on this task is inferior to performance of single-speaker detection so the scale in Figure 4 has been expanded relative to Figure 3. The improvements gained over the last year can be seen as the EER improved from 16.5% for last year's system to 14.0% for the current system. The figure also shows a small performance gain from combining the internal (SRE09) and external (SRE00) seg-
mentation systems. To demonstrate the performance improvement gained by segmenting the speech into speaker homogeneous regions, the performance of the one-speaker detection system on the two-speaker detection task is also shown in Figure 4.

The results for two-speaker tracking are shown in Figure 5. The plot compares last year's system, an internal segmentation system, with this year's external segmentation system. Also shown is a combination of the two systems. The plot shows that the external segmentation system outperforms the internal segmentation system at most operating points. Combining the two systems yields a significant performance gain at low false alarm rates.

Results for the speaker segmentation tasks are shown in Tables 1 and 2. The results for the case in which the number of speakers is known to be two appear in Table 1. The overall error is 0.10 but the breakdown of the errors shows that performance is slightly better for mixed gender conversations and when both talkers are male, but performance is significantly worse when both talkers are female.

Table 1: Two-speaker segmentation results.

<table>
<thead>
<tr>
<th>Condition</th>
<th># Files</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Files</td>
<td>1000</td>
<td>0.10</td>
</tr>
<tr>
<td>2 male speakers</td>
<td>274</td>
<td>0.08</td>
</tr>
<tr>
<td>2 female speakers</td>
<td>321</td>
<td>0.17</td>
</tr>
<tr>
<td>Mixed Gender speakers</td>
<td>408</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Table 2: N-speaker segmentation results.

<table>
<thead>
<tr>
<th>Condition</th>
<th># Files</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Files</td>
<td>1000</td>
<td>0.20</td>
</tr>
<tr>
<td>2-speakers</td>
<td>303</td>
<td>0.13</td>
</tr>
<tr>
<td>3-speakers</td>
<td>106</td>
<td>0.24</td>
</tr>
<tr>
<td>4-speakers</td>
<td>101</td>
<td>0.22</td>
</tr>
<tr>
<td>5-speakers</td>
<td>10</td>
<td>0.32</td>
</tr>
<tr>
<td>6-speakers</td>
<td>6</td>
<td>0.38</td>
</tr>
<tr>
<td>7-speakers</td>
<td>2</td>
<td>0.40</td>
</tr>
<tr>
<td>arabic</td>
<td>55</td>
<td>0.20</td>
</tr>
<tr>
<td>english</td>
<td>55</td>
<td>0.14</td>
</tr>
<tr>
<td>german</td>
<td>15</td>
<td>0.14</td>
</tr>
<tr>
<td>japanese</td>
<td>68</td>
<td>0.22</td>
</tr>
<tr>
<td>mandarin</td>
<td>118</td>
<td>0.23</td>
</tr>
<tr>
<td>spanish</td>
<td>56</td>
<td>0.22</td>
</tr>
</tbody>
</table>

5 CONCLUSIONS AND FUTURE WORK

In this paper we describe the Lincoln systems for SRE00. The system was the best performing system in the evaluation for two-speaker detection, two-speaker tracking, and N-speaker segmentation. In the other tasks the performance of the Lincoln system was comparable to the best performing system. This demonstrates that the adopted GMM-HMM system is one of the most effective systems for text-independent speaker detection. We demonstrate tools which are combine with the onespeaker detection system to handle multi-speaker detection.

Future work is required to improve performance on multiprobe tasks. Segmentation of an unknown number of speakers remains a difficult task, and even when only two speakers are present, the error is 0.13. There is also significant room for improving text-independent onespeaker detection on telephone speech, where channel normalization presents a significant challenge. The equal error rate of the best systems on this difficult data is near 10%.

REFERENCES


