Recent Improvements in the CUED Diarisation System

Sue Tranter, Rohit Sinha, Mark Gales & Phil Woodland

March 19th 2005

Cambridge University Engineering Department
### Progress Since RT-04 Workshop

<table>
<thead>
<tr>
<th>System</th>
<th>Dev-24 DER</th>
<th>Dev-12 DER</th>
<th>Eval DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT-04 Eval System (Oct 2004)</td>
<td>17.7%</td>
<td>17.2%</td>
<td>24.0%</td>
</tr>
<tr>
<td>RT-04 Workshop (Nov 2004) (without topdown clustering)</td>
<td>19.2%</td>
<td>20.2%</td>
<td>17.9%</td>
</tr>
<tr>
<td>MDE Tech Meeting (Mar 2005)</td>
<td>9.0%</td>
<td>7.7%</td>
<td>6.9%</td>
</tr>
</tbody>
</table>

Dev-24 data = 24 shows (eval03, didev03, dev04f2, sttdev04)
Dev-12 data = 12 shows (eval03, dev04f2 = RT-04 diarisation dev data)
Eval data = 12 shows (eval04 - reference 22nd Dec 2004)
System Architecture - RT04 Workshop

- MFCC/PLP coding
- Speech/Music/Narrowband GMMs
- Bandwidth labels
- Phone recognition
- Inter-silence segments
- Divergence-based change detection
- Over-segmented data
- Build model for each segment
- Viterbi Segmentation and re-estimation
- Threshold-based clustering
- no change or max iterations

CPD

IACS
Iterative Agglomerative Clustering Stage (IACS)

- Run in two stages, first with diagonal (PLP_0_D_A) and second with full covariance (PLP_0).
- Each stage runs up to 6 iterations.
- RT-04 system used a constant threshold on the likelihood for merge decisions (no BIC penalty weight).
- The method of updating when clusters were combined was changed from centroid clustering to forming the stats from the concatenated data.
- Options for using a (‘local’) BIC criterion for ordering the merges and/or merge decision were added.
- A furthest neighbour scheme (which didn’t need distance recomputation after each merge) was also added.
IACS - Distance Metrics

The diagonal covariance step was run conservatively to oversegment the data, and fixed for these experiments on the full covariance stage.

<table>
<thead>
<tr>
<th>ID</th>
<th>Clustering</th>
<th>Ordering</th>
<th>Decision</th>
<th>Opt-dev</th>
<th>Eval</th>
<th>(Opt Eval)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Centroid</td>
<td>0</td>
<td>constant</td>
<td>19.4</td>
<td>17.7</td>
<td>(17.6)</td>
</tr>
<tr>
<td>2</td>
<td>Concat.</td>
<td>0</td>
<td>constant</td>
<td>19.1</td>
<td>17.1</td>
<td>(17.1)</td>
</tr>
<tr>
<td>3</td>
<td>Concat.</td>
<td>0</td>
<td>BIC</td>
<td>18.9</td>
<td>20.3</td>
<td>(16.8)</td>
</tr>
<tr>
<td>4</td>
<td>Concat.</td>
<td>BIC</td>
<td>BIC</td>
<td>18.6</td>
<td>17.9</td>
<td>(17.7)</td>
</tr>
<tr>
<td>5</td>
<td>Furthest N</td>
<td>0</td>
<td>constant</td>
<td>18.5</td>
<td>19.3</td>
<td>(19.0)</td>
</tr>
</tbody>
</table>

- Furthest Neighbour (5) performed best on the dev but not the eval data.
- Using a constant in the decision (2) gave the best (non-tuned) eval score.
- The more standard BIC method (4) did reasonably on both data sets.
- The results are often sensitive to relatively small parameter changes.
IACS - Summary

The baseline system uses:

- ‘local’ BIC for both ordering and decision in merging stage
- Multiple iterations at optimal (dev) $\alpha$
- Phasing of 1 iteration of $\alpha = 1$.
- This underclusters the data for subsequent SID stage.

The results are:

<table>
<thead>
<tr>
<th>DataSet</th>
<th>MS/FA/SPE/DER</th>
<th>Cluster Imp †</th>
<th>Seg Imp †</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev-24</td>
<td>1.2/1.1/17.9/20.17</td>
<td>6.59 @ 718</td>
<td>4.36 @ 2363</td>
</tr>
<tr>
<td>Dev-12</td>
<td>1.0/1.3/20.2/22.47</td>
<td>5.32 @ 321</td>
<td>4.22 @ 1078</td>
</tr>
<tr>
<td>Eval (12)</td>
<td>0.3/1.1/17.4/18.75</td>
<td>5.04 @ 336</td>
<td>3.63 @ 1072</td>
</tr>
</tbody>
</table>

† Seg/Cluster Imp = DER with oracle clustering of segments/clusters. (including MS/FA)
System Architecture - Adding SID

- MFCC/PLP coding
- Speech/Music/Narrowband GMMs
- Bandwidth labels
- Phone recognition
- Inter-silence segments

Divergence-based change detection

Over-segmented data

- Build model for each segment
- Viterbi Segmentation and re-estimation
- Threshold-based clustering

no change or max iterations

Empty Segs

P1 STT and Gender Labelling

Feature Warping

SID: GD Map adaptation to UBM

Final Output

CPD

IACS

GD

SID
SID stage - Description

- Based on LIMSI’s "SID-like" stage in their RT-04 evaluation system.
- Perform agglomerative speaker clustering using the cross log-likelihood ratio (CLR) between clusters.
- Clustering is done separately for each bandwidth and gender.
- Each cluster model is derived by MAP adapting (mean only) a universal background model (UBM).
- The stopping criterion used is a threshold on global CLR, $\theta_{CLR}$.
- Feature warping is applied to reduce the effect of acoustic environment.
SID stage - Effect of Features and Feature Warping

Different warped features were investigated in particular the inclusion of c0 (\_0), energy (\_E), or just the differentials thereof (\_N).

- PLP with deltas and no energy performed the best.
- Feature warping improved the DER from 18.1% to 10.8%.
SID stage - Variable Prior (VP) MAP

For MAP adaptation the mean, $\mu$, is changed depending on a prior model $p$ and the data $d$:

$$\hat{\mu}^{(1)} = \frac{\gamma_d^{(1)} \mu_d^{(1)} + \tau \mu_p}{\gamma_d^{(1)} + \tau}$$

- A small $\tau$ makes the mean stick to a few speakers in the data and thus is robust to the SID threshold, $\theta_{CLR}$, but may get 'misled' by the data.

Variable Prior (VP) MAP uses $\hat{\mu}^{(N)}$ instead of $\mu_p$ for iteration $N+1$.

e.g. 2nd iteration

$$\hat{\mu}^{(2)} = \frac{\gamma_d^{(2)} \mu_d^{(2)} + \tau \left( \frac{\gamma_d^{(1)} \mu_d^{(1)} + \tau \mu_p}{\gamma_d^{(1)} + \tau} \right)}{\gamma_d^{(2)} + \tau}$$

- The numerator $\mu_p$ term becomes weighted by $(\frac{\tau^2}{\gamma_d^{(1)} + \tau}) \leq \tau$
- More iterations decreases the prior's influence as the new models improve.
SID stage - Type of MAP and $\tau$

We compared MAP and VP-MAP for different $\tau$ values and 2 iterations.

- VP-MAP outperforms MAP for every value of $\tau$.
- Use VP-MAP, $\tau = 10$, giving 10.8% on dev and 9.9% on eval. (opteval=9.2%)
• Use 2 iterations of VP-MAP
• This gives 10.8% on dev and 9.9% on eval.
**SID stage - Narrowband Data**

How to deal with the automatically labelled NB data in the SID stage:

- **NB-none** Pass NB clusters directly to output (default)
- **NB-WB** Run NB clusters through SID using WB coding
- **NB-NB** Run NB clusters through SID using NB coding

![Diagram showing Dev Data vs SID Threshold]

- Use NB coding for NB clusters. (Using WB makes worse - add a NB $\theta$ ?)
- Hardly any data classified as NB for eval04 makes this less worthwhile.
- This gives 10.7% on dev and 9.9% on eval.
SID stage - UBM generation

Different UBMs were built for experiments with 256, 512 and 1024 mixtures.

**Orig** 6 hrs per gender taken from hub4-train 96/7

**Bal** 7.5 hrs per gender taken evenly across all sources in hub4-train 96/7

- The Balanced set performed best with 1024 mixtures and $\theta_{CLR} = 0.3$
- This gives 10.4% on dev and 9.9% on eval. (opteval = 8.9%)
SID stage - Adding Dev Data to UBM

- sttdev04 and didev03 development data, $D$, included in UBM.
- B+D retrained UBM with Bal+D data, BMAPD MAPed Bal to D
- dev-12 remained 'uncontaminated' dev set.

- Using dev data made much larger improvements on eval than dev-12 DER.
  - BMAPD-1024 gives 8.5% on dev-12 and 10.4% on eval (opteval=8.0%)
  - B+D-1024, gives 8.4% on dev-12 and 9.5% on eval (opteval=8.3%)
SID stage - Adding Eval Data to UBM

- New UBMs, B+E and BMAPE, were created using the whole test data set.
- System gender labels were used. (no real cheating but against eval rules).
- Using just the target show (rather than whole test set) did not work.

- Using all the test data improved the best performance over the Bal UBM.
- BMAPE-1024, gives 10.2% on dev24 and 7.8% on eval (opteval 7.8%).
- B+E-1024 gives 9.9% on dev24 and 8.3% on eval (opteval 8.3%).
SID stage - Summary

• We built a successful SID-like stage using LIMSI’s as a base model.

• Feature warping slashed our dev24 DER from 18.1% to 10.8%.

• VP-MAP was introduced and shown to outperform MAP.

• 2 iterations of VP-MAP using PLP_D and $\tau = 10$ worked best.

• Carefully adding (reference) dev data into the UBM helped eval performance.

• The final system gave a DER of 8.4% on dev12 and 9.5% on eval.

• Adding the test data itself into the UBM improved performance, giving 8.3% or 7.8% on the eval data depending on the method used.
SID stage - Using LIMSI’s Segments

LIMSI were kind enough to provide us with the input they used for their SID stage in the RT-04 evaluation.

<table>
<thead>
<tr>
<th>SID input</th>
<th>Segment Impurity</th>
<th>Cluster Impurity</th>
<th>DER</th>
<th>SID DER*</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUED</td>
<td>MS/FA/SPE/DER @ #Seg</td>
<td>DER @ #Spk</td>
<td></td>
<td>θ_{dev} (θ_{eval})</td>
</tr>
<tr>
<td>LIMSI</td>
<td>0.2/1.8/1.0/3.05 @ 1110</td>
<td>4.02 @ 477</td>
<td>18.4</td>
<td>9.1 (7.6)</td>
</tr>
<tr>
<td></td>
<td>0.3/1.1/2.3/3.63 @ 1072</td>
<td>5.04 @ 336</td>
<td>18.8</td>
<td>9.5 (8.3)</td>
</tr>
</tbody>
</table>

* B+D-1024 model used, τ = 10, VP-MAP, 2 iterations

- DERs of 7% were obtained using LIMSI’s SID input with different UBM models. (With a further gain of 0.6% by using the CUED SAD labels)
- We need to improve our ‘pre-SID’ segmentation/clustering!
Altering the Change Point Detection

The change point detection was rewritten:

- Finding peaks directly from distance metric improved potential purity.
- New minimum length constraint enforced by removing the smaller of neighbouring peaks (L to R) reduced number of segments dramatically.

Results:

- Larger window size improved performance.
- Full covariance model worked better than diagonal for the larger window size.
- Switching features from PLP_0_D to MFCC_E_D_A_N did not help.
- Using feature warping degraded performance severely.
Change Point Detection - Results

The speaker error component from the ideal clustering of the segments is used to measure the segment impurity.

- Best Performance from 2s windows, 1s min length, full covariance.
### Change Point Detection - Effect on Whole System

<table>
<thead>
<tr>
<th>dev-24 data</th>
<th>CPD out Seg Imp†</th>
<th>IACS out Clust Imp†</th>
<th>MS/FA/DER</th>
<th>dev-12* SID DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>3.99 @10573</td>
<td>4.36 @ 2363</td>
<td>6.59 @718</td>
<td>1.2/1.1/20.2</td>
</tr>
<tr>
<td>newCPD-diagc</td>
<td>2.59 @11348</td>
<td>4.11 @ 2385</td>
<td>6.40 @722</td>
<td>1.2/1.1/19.2</td>
</tr>
<tr>
<td>newCPD-fullc</td>
<td>2.50 @11299</td>
<td>4.21 @ 2371</td>
<td>6.39 @720</td>
<td>1.2/1.1/20.3</td>
</tr>
</tbody>
</table>

* B+D-1024 model used, $\tau = 10$, VP-MAP, 2 iterations, $\theta_{opt}(dev12)$
† Segment/Cluster Imp = DER with oracle clustering of segments/clusters. (including MS/FA)

- Segment purity much better after CPD stage (~0.25% SPE).
- Early promise not carried through. (no gain in DER seen on eval data)
- IACS should be adjusted (e.g. removing diag cov stage as segs now >1s).
- First results on retuned IACS give 7.4/8.8% on dev 12/24 and 8.6% on eval. (Results with B+E models give 7.7/9.0% on dev12/24 and 6.9% on eval.)
Future Work

Short Term Goals

• Re-tune IACS with new CPD output and try with B+E model.

Things to think about

• CPD: add smoothing such as a median filter or hamming window

• IACS: Try incorporating feature warping

• SID: Use BW-labelled data to build GMMs.

• FINAL: Post-process output with STT cues.
Conclusions

- The DER of the CUED Diarisation system on the eval04 data has been reduced from 17.9% to 6.9% with a similar drop in the dev data DER.

- Most of the improvement came from adding a SID-like stage in a similar style to LIMSI’s.

- DERs of around 6.5% are possible on eval04 data with this method.

- These experiments will be written up for a Eurospeech 2005 submission.

Thanks are due to the LIMSI speaker recognition team, and in particular Claude Barras for helping us improve our system performance by providing intermediate files and helpful advice.