Progress in English Conversational Telephone Speech Transcription

Khe Chai Sim, Mark Gales, Xunying Liu, Phil Woodland & Kai Yu

March 2005

Cambridge University Engineering Department
Outline

- Increased number of model parameters
  - 9K states to 15K states
  - use of multiple STCs

- Combination results

- Initial experiments with fMPE
  - view fMPE as temporally varying shift of mean vectors
  - extension: pMPE as temporally varying scale of precision matrices
Acoustic Training Set-Up

- Acoustic Model Training Data (fsh2004h5train03b - 2180hours):
  - h5train03b: 360hours used in 2003 evaluation
  - fsh2004: 1820hours BBN/Wordwave+LDC quick transcriptions

- Acoustic Model Test Data:
  - eval03 6 hours (3 hours Switchboard2 Phase 5, 3 hours Fisher)
  - dev04 3 hours Fisher data

- Front-end
  - 12 PLP cepstral parameters + C0 and 1st/2nd/3rd derivatives + HLDA
  - Side-based cepstral mean and variance normalisation plus VTLN

- Baseline Acoustic Models
  - Gender independent, decision tree state clustered triphones
  - MPE training with dynamic MMI prior

- Language Models (see RT04f workshop paper for details)
  - 2003 trigram (tgint03) unadapted decodes
  - 2004 evaluation LM for 10xRT experiments
### Varying Model Complexity

<table>
<thead>
<tr>
<th>System</th>
<th># States (# Comp)</th>
<th>MPE Iter</th>
<th>eva03</th>
<th>dev04</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>6K (28)</td>
<td>0 (ML)</td>
<td>34.1</td>
<td>26.4</td>
</tr>
<tr>
<td>S4</td>
<td>9K (36)</td>
<td></td>
<td>33.0</td>
<td>25.3</td>
</tr>
<tr>
<td>S6</td>
<td>15K (36)</td>
<td></td>
<td>32.1</td>
<td>24.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th># States (# Comp)</th>
<th>MPE Iter</th>
<th>eva03</th>
<th>dev04</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>6K (28)</td>
<td>8</td>
<td>27.9</td>
<td>20.5</td>
</tr>
<tr>
<td>S4</td>
<td>9K (36)</td>
<td></td>
<td>26.8</td>
<td>20.0</td>
</tr>
<tr>
<td>S6</td>
<td>15K (36)</td>
<td></td>
<td>26.5</td>
<td>19.7</td>
</tr>
</tbody>
</table>

%WER GI unadapted decode 2003 trigram

- S6 system (540K) is $1.7 \times$ larger than S4 system (320K) on dev04
  - S6 MLE 1.0%/MPE 0.3% better than S4 MLE/MPE system
- Unfortunately MPE gains consistently less than MLE gains
  - complexity affects MPE gains - on dev04

<table>
<thead>
<tr>
<th>System</th>
<th>%Gains</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>5.9%</td>
</tr>
<tr>
<td>S4</td>
<td>5.3%</td>
</tr>
<tr>
<td>S6</td>
<td>4.6%</td>
</tr>
</tbody>
</table>
10xRT Framework

- Evaluation 10xRT framework:
- Multi-pass framework
- Confusion network generation
- Confusion network combination
- Evaluation system used:
  - P3b: S4: Triphone GD MPron
  - P3q: Q1: Quinphone SAT SPron
- Use alternative P3 branches
  - ignore time constraints ...
### 10xRT Framework Results

<table>
<thead>
<tr>
<th>System</th>
<th>eval03</th>
<th>dev04</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s25</td>
<td>fsh</td>
</tr>
<tr>
<td>P3b-cn</td>
<td>21.7</td>
<td>14.7</td>
</tr>
<tr>
<td>P3q-cn</td>
<td>21.5</td>
<td>14.8</td>
</tr>
<tr>
<td>P3d-cn</td>
<td>21.3</td>
<td>14.5</td>
</tr>
<tr>
<td>P3s-cn</td>
<td>21.0</td>
<td>14.6</td>
</tr>
<tr>
<td>P3b+P3q</td>
<td>20.9</td>
<td>14.1</td>
</tr>
<tr>
<td>P3b+P3d</td>
<td>21.3</td>
<td>14.3</td>
</tr>
<tr>
<td>P3d+P3q</td>
<td>20.6</td>
<td>13.9</td>
</tr>
<tr>
<td>P3s+P3q</td>
<td>20.4</td>
<td>14.0</td>
</tr>
<tr>
<td>P3s+P3d</td>
<td>20.8</td>
<td>14.3</td>
</tr>
</tbody>
</table>

% WER 2004 10xRT rescoring/combination, 2004 RT04f LMs

- Best single branch S4 SAT-SPAM system (too slow for real 10xRT!)
- S6 GD about 0.2% better than S4 GD
- Gain maintained after combination with Q1, 0.2% better than eval system.
Semi-Tied Covariance Matrices (reminder)

- IBM investigated full covariance matrices
  - simpler updates than SPAM/EMLLT systems
  - but dramatic increase in number of parameters/decode cost
  - necessary to limit number of components (IBM: 144K vs 800K)
- Examine simpler precision matrix models - semi-tied covariance matrices
  - simple/efficient update formulae
  - efficient likelihood calculation

\[
L(o; \mu^{(m)}, \Sigma^{(m)}_{\text{diag}}, A^{(r)}) = |\det(A^{(r)})|N(A^{(r)}o; \mu^{(m)}, \Sigma^{(m)}_{\text{diag}})
\]

- HLDA subsumes a global STC transform
- Normally only a small number of semi-tied transforms considered
  - with more data, dramatically increase the number of transforms
  - no need to limit number of components
## Unadapted STC Results

<table>
<thead>
<tr>
<th>System</th>
<th>#STC XForms</th>
<th>MPE Iter</th>
<th>eval03</th>
<th>dev04</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>s25</td>
<td>fsh</td>
</tr>
<tr>
<td>S4</td>
<td>—</td>
<td>0</td>
<td>31.6</td>
<td>24.3</td>
</tr>
<tr>
<td></td>
<td>1K (ML)</td>
<td></td>
<td>31.3</td>
<td>24.0</td>
</tr>
<tr>
<td></td>
<td>9K</td>
<td></td>
<td>30.7</td>
<td>23.1</td>
</tr>
<tr>
<td>S4</td>
<td>—</td>
<td>8</td>
<td>26.7</td>
<td>19.6</td>
</tr>
<tr>
<td></td>
<td>9K</td>
<td></td>
<td>26.3</td>
<td>18.9</td>
</tr>
<tr>
<td>S6</td>
<td>—</td>
<td>8</td>
<td>26.5</td>
<td>19.4</td>
</tr>
</tbody>
</table>

%WER GI unadapted decode, HDecode, PronProbs, 2003 trigram

- S4 9K STC system is 1.6× larger than standard S4 system
  - S4 9K STC system MLE 1.1%/MPE 0.6% absolute better than S4 system
  - slightly better (0.1%-0.3%) than S6 system
- Unfortunately adaptation more complex (same as full cov)
fMPE – from the Model Parameter Point of View

• IBM’s fMPE — a form of feature interpolation based on posterior information

• Equivalent to temporally varying *shift* of mean vectors

\[
\mu_{mt} = \mu_m + \sum_{i=1}^{n} p(c_i|o_t)b_i = \mu_m + b_t
\]

– \(c_i\): cluster centroid, \(n \gg d\)

• *Static* (\(\mu_m\)) & *dynamic* (\(b_t\)) parameters

• Interleaving update of static and dynamic parameters:
  - **static**: update \(\mu_m\) (ML); fix \(b_i\)
  - **dynamic**: update \(b_i\) (fMPE); fix \(\mu_m\)
Update of Temporal Mean Vectors

- Update of $\mu_m$ (ML):
  $$\mu_m = \frac{\sum_{t=1}^{T} \gamma_m(t)(o_t-b_t)}{\sum_{t=1}^{T} \gamma_m(t)}$$

- Update of $b_{ij}$ (fMPE):
  $$\hat{b}_{ij} = b_{ij} + \eta_{ij} \frac{\partial F}{\partial b_{ij}}$$

\[
\frac{\partial F}{\partial b_{ij}} = \sum_{t=1}^{T} p(c_i|o_t) \left\{ \sum_{m=1}^{M} \left( \frac{\partial F}{\partial L^m} \frac{\partial L^m}{\partial \mu_{mjt}} + \frac{\partial F}{\partial \mu_{mj}} \frac{\partial \mu_{mjt}}{\partial \mu_{mjt}} + \frac{\partial F}{\partial \sigma^2_{mj}} \frac{\partial \mu_{mjt}}{\partial \mu_{mjt}} \right) \right\}
\]

\[
L^m = K - \frac{1}{2} \sum_{j=1}^{d} \left( \log(\sigma^2_{mj}) - \frac{(o_{jt} - \mu_{mjt})^2}{\sigma^2_{mj}} \right)
\]

- Exactly the same as fMPE ...

- But, motivates the extension for *temporal precision matrices*
Temporal Precision Matrices (pMPE)

- Temporal scaling of diagonal precision elements \((s_{mj} = 1/\sigma_{mj}^2)\)

\[
s_{mt} = \left(1 + \sum_{i=1}^{n} p(c_i|o_t)a_{ij}\right)^2 s_{mj} = a_{jt}^2 s_{mj}
\]

- Positive temporal scaling, \(a_{jt}^2\), to ensure positive variances
- Similar interleaving update as fMPE
- Likelihood calculation more expensive:

\[
\mathcal{L}^m = K + \frac{1}{2} \sum_{j=1}^{d} \left( \log(s_{mj}) + \log(a_{jt}^2) - a_{jt}^2 s_{mj}(o_{jt} - \mu_{mj})^2 \right)
\]

- cache \(a_{jt}^2\) and \(\sum_{j=1}^{d} \log(a_{jt}^2)\)
- extra \(d\) multiplications and 1 addition
Update of Temporal Precision Matrices

• Update of $\sigma^2_{m,j}$ (ML):

$$\sigma^2_{m,j} = \frac{\sum_{t=1}^{T} \gamma_m(t) a_{jt}^2 (o_{jt} - \mu_{mjt})^2}{\sum_{t=1}^{T} \gamma_m(t)}$$

• Update of $a_{ij}$ (pMPE):

$$\hat{a}_{ij} = a_{ij} + \eta_{ij} \frac{\partial F}{\partial a_{ij}}$$

$$\frac{\partial F}{\partial a_{ij}} = 2 \sum_{t=1}^{T} a_{jt} p(c_i | o_t) \left\{ \sum_{m=1}^{M} \left( \frac{\partial F}{\partial L^m_m} \frac{\partial L^m_m}{\partial s_{mjt}} + \frac{\partial F}{\partial \sigma^2_{m,j}} \frac{\partial \sigma^2_{m,j}}{\partial s_{mjt}} + \frac{\partial F}{\partial \mu_{mjt}} \frac{\partial \mu_{mjt}}{\partial s_{mjt}} \right) \right\}$$

• Learning rate:

$$\eta_{ij} = \frac{\alpha}{(p_{ij} + n_{ij})}$$
Experimental Setup

- Acoustic model data sets:
  - **Training data**: 76 hours h5ettrain03sub & 296 hours h5ettrain03
  - **Test data**: 3 hours dev01sub & 6 hours eval03
- Front-end: Standard CUED CTS set-up
- Posterior calculations:
  - $\sim 70k$ & $\sim 100k$ Gaussians for posterior calculation
  - Gaussians grouped into 1024 clusters
  - Evaluate the top 5 Gaussians with $\sim 2$ active posteriors/frame
  - Single frame posteriors *without* context
- Baseline Acoustic Models
  - 12 & 16 component VarMix gender independent
  - Decision tree state clustered triphones ($\sim 6000$ states)
  - MPE training *without* dynamic MMI prior
- Trigram Language Models
dev01sub results of fMPE trained on h5etrain03sub

- fMPE (4 iter): 32.7% (+1.6% over ML)
- fMPE+MPE (8 iter): 31.1% (+1.4% over MPE & +3.2% over ML)
- pMPE & pMPE+MPE: less robust to overtraining
**dev01sub results of fMPE & pMPE trained on h5etrain03**

- fMPE & fMPE+MPE: similar gain as before
- pMPE converged quicker (≈ 2 iterations) with ≈ 1.0% gain over ML
- pMPE+MPE gave 0.2-0.3% gain over MPE alone
- fpMPE further 0.5% gain over fMPE
**eval03 results of fMPE & pMPE trained on h5etrain03**

<table>
<thead>
<tr>
<th>System</th>
<th>Iter 0</th>
<th></th>
<th></th>
<th></th>
<th>Iter 8</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>s25</td>
<td>fsh</td>
<td>Avg</td>
<td>s25</td>
<td>fsh</td>
<td>Avg</td>
<td>s25</td>
</tr>
<tr>
<td>MPE</td>
<td>36.4</td>
<td>27.1</td>
<td>31.9</td>
<td></td>
<td>33.6</td>
<td>24.2</td>
<td>29.1</td>
<td>33.2</td>
</tr>
<tr>
<td>fMPE+MPE</td>
<td>34.5</td>
<td>25.4</td>
<td>30.1</td>
<td></td>
<td>32.7</td>
<td>23.3</td>
<td>28.1</td>
<td>32.3</td>
</tr>
<tr>
<td>pMPE+MPE</td>
<td>35.1</td>
<td>26.0</td>
<td>30.7</td>
<td></td>
<td>33.2</td>
<td>24.1</td>
<td>28.8</td>
<td>32.9</td>
</tr>
</tbody>
</table>

%WER of 16-component systems on eval03

- Performance of fMPE and pMPE on eval03 similar to dev01sub
- Gains over ML: fMPE (1.8%), pMPE (1.2%), fpMPE (2.0%)
- Improvement over MPE alone: fMPE+MPE (0.8-1.0%), pMPE+MPE (0.2-0.3%) and fpMPE+MPE (0.4-0.5%)
Summary

- S6 (15k states) gave $\sim 0.2-0.3\%$ absolute gain

- STC 9K system:
  - alternative approach to building full covariance matrix system
  - gave $\sim 0.6\%$ absolute gain (unadapted MPE)

- Initial fMPE results – similar gains to IBM

- Gains from pMPE smaller compared to fMPE

- Future work:
  - apply fMPE to larger training set (fsh2004h5etrain03b)
  - investigate interaction between fMPE and pMPE
  - lattice regeneration for fMPE+MPE training