Tandem modeling investigations

Dan Ellis
International Computer Science Institute, Berkeley CA
<dpwe@icsi.berkeley.edu>

Outline

1. What makes Tandem successful?
2. Can we make Tandem better?
3. Does Tandem work with LVCSR tricks?
What makes Tandem work?
(with Manuel Reyes)

- Model diversity?
  - try a phone-based GMM model
  - try training the NN model to HTK state labels

- Discriminative network training?
  - (try posteriors from GMM & Bayes)
Phone vs. word models

- Try a phone-based HTK model (instead of whole-word models)
- Try training NN model to subword-state labels
  - 181 net outputs; reduce to 40 in KLT
- Results (Aurora2k, HTK-baseline WER ratio):

<table>
<thead>
<tr>
<th>System</th>
<th>test A: matched</th>
<th>test B: var noise</th>
<th>test C: var chan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tandem PLP baseline</td>
<td>63.5%</td>
<td>70.3%</td>
<td>59.5%</td>
</tr>
<tr>
<td>Phone-based HTK sys</td>
<td>63.6%</td>
<td>72.5%</td>
<td>61.5%</td>
</tr>
<tr>
<td>Subword-based NN sys</td>
<td>63.1%</td>
<td>62.8%</td>
<td>55.1%</td>
</tr>
</tbody>
</table>

- Diversity doesn’t help
  - subword units may be good for NN
2 Enhancements to Tandem-Aurora

- More tandem-feature-domain processing:

- Results (HTK baseline WER ratio):

<table>
<thead>
<tr>
<th>System</th>
<th>test A: matched</th>
<th>test B: var noise</th>
<th>test C: var chan</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP: Tandem baseline</td>
<td>63.5%</td>
<td>70.3%</td>
<td>59.5%</td>
</tr>
<tr>
<td>PLP: norm - KLT</td>
<td>72.6%</td>
<td>71.2%</td>
<td>63.6%</td>
</tr>
<tr>
<td>PLP: KLT - norm</td>
<td>57.8%</td>
<td>58.8%</td>
<td>61.3%</td>
</tr>
<tr>
<td>PLP: KLT - delta</td>
<td>59.0%</td>
<td>60.2%</td>
<td>52.9%</td>
</tr>
<tr>
<td>PLP: KLT - delta - norm</td>
<td>58.1%</td>
<td>59.9%</td>
<td>48.9%</td>
</tr>
<tr>
<td>PLP: delta - KLT - norm</td>
<td>54.7%</td>
<td>53.6%</td>
<td>46.9%</td>
</tr>
</tbody>
</table>

- delta-KLT-norm: 80% Tdm baseline WER
Best effort Tandem system

- **Deltas & norms help PLP:**
  try on combo (PLP+MSG) system:

<table>
<thead>
<tr>
<th>System</th>
<th>test A: matched</th>
<th>test B: var noise</th>
<th>test C: var chan</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP+MSG: baseline</td>
<td>51.1%</td>
<td>52.0%</td>
<td>45.6%</td>
</tr>
<tr>
<td>PLP+MSG: dlt-KLT-nrm</td>
<td>50.9%</td>
<td>50.5%</td>
<td>43.6%</td>
</tr>
<tr>
<td>PLP+MSG: KLT-nrm</td>
<td>48.3%</td>
<td>49.5%</td>
<td>39.4%</td>
</tr>
</tbody>
</table>

- deltas *hurt* for MSG: features too sluggish?

- **Deltas help clean, norms help noisy:**

![WER vs SNR graph](image-url)

SNR / dB

WER / %

WER / %

baseline

K-D

K-N
### Tandem for LVCSR: the SPINE task

(with Rita Singh/CMU & Sunil Sivadas/OGI)

- **Noisy spontaneous speech, ~5000 word vocab**

- **Recognition:**
  - same tandem features
  - NN training from Broadcast News boot + iterate
  - GMM-HMM has context-dependence, MLLR
SPINE-Tandem results

- Evaluation WER results:

<table>
<thead>
<tr>
<th>Features (dimensions)</th>
<th>CI system</th>
<th>CD system</th>
<th>CD + MLLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC + d + dd (39)</td>
<td>69.5%</td>
<td>35.1%</td>
<td>33.5%</td>
</tr>
<tr>
<td>Tandem features (56)</td>
<td>47.6%</td>
<td>35.7%</td>
<td>32.8%</td>
</tr>
</tbody>
</table>

- much better for CI systems
- differences evaporate with CD, MLLR

- Not quite fair:
  - CD senones optimized for MFCC
  - worth 2-3% absolute?

- Not unexpected:
  - NN confounds CD variants
  - Tandem ‘space’ very nonlinear - bad for MLLR

- Any hope?
  - more training data / train CD classes / ...