SRI 2001 SPINE Evaluation System

Venkata Ramana Rao Gadde
Andreas Stolcke
Dimitra Vergyri
Jing Zheng
Kemal Sonmez
Anand Venkataraman
Talk Overview

- System Description
  - Components
    - Segmentation
    - Features
    - Acoustic models
    - Acoustic Adaptation
    - Language models
    - Word posteriors
    - System Combination
  - Processing Steps
- Results
  - Dryrun, Evaluation
  - What worked
  - What didn't
  - Fourier cepstrum revisited
- Evaluation Issues
- Future Work
- Conclusions
System Description
Segmentation

- Segmentation is done in multiple steps
  - Classify and segment waveform into foreground/background using a 2-class HMM
  - Recognize foreground segments
  - Compute word posterior probabilities (from confusion networks derived from N-best lists)
  - Resegment the foreground segments eliminating word hypotheses with posteriors below a threshold (optimized on dryrun data)
Acoustic Features

- 3 feature streams with separate acoustic models:
  - Mel cepstrum
  - PLP cepstrum (implementation from ICSI)
  - Fourier cepstrum

- Each feature stream has 39 dimensions consisting of 13 cepstra, 13 deltas and 13 delta-deltas

- Features were normalized for each speaker
  - Cepstral mean and variance normalization
  - Vocal tract length normalization
  - By transforms estimated using constrained MLLR
Acoustic Models

- 6 different acoustic models:
  - 3 frontends
  - crossword + non-crossword
- All models gender-independent
- SPINE1 training + eval + SPINE2 training data
- Bottom-up clustered triphone states ("genones")
- Non-crossword models contained about 1000 genones with 32 gaussians/genones
- Crossword models contained about 1400 genones with 32 gaussians/genones
Discriminative Acoustic Training

- All models were first trained using the standard maximum likelihood (ML) training
- Subsequently, one additional iteration of discriminative training, using maximum mutual information estimation (MMIE)
Acoustic Adaptation

- Adaptation was applied in two different ways
  - Feature normalization using constrained MLLR
    - Feature normalization transforms were computed using a reference model, trained from VTL and cepstral mean and variance normalized data.
    - A global model transform was computed using the constrained MLLR algorithm and its inverse was used as the feature transform.
    - Equivalent to speaker-adaptive training (Jin et al, 1998).
Acoustic Adaptation (continued)

- Model adaptation using modified MLLR
  - Acoustic models were adapted using a variant of MLLR which does variance scaling in addition to mean transformation.
  - 7 phone classes were used to compute the transforms.
Language Models

- 3 language models (**4 evaluation systems**):
  - SRI LM1: trained on SPINE1 training + eval data, SPINE2 training + dry run data (**SRI1, SRI2**)
  - SRI LM2: trained on SPINE1 training + eval data, SPINE2 training data (**SRI3**)
  - CMU LM: modified to include multiword n-grams (**SRI4**)

- Trigrams used in decoding, 4-grams in rescoring.

  **Note:** SRI4 had bug in LM conversion.
  - Official result: 42.1% Corrected result: 36.5%.
Class-based Language Model

- Goal: Overcome mismatch between 2000 and 2001 task vocabulary (new grid vocabulary)
- Approach (similar to CU and IBM):
  - Map 2000 and 2001 grid vocabulary to word classes
  - 2 classes: grid words and spelled grid words
  - Expand word classes with uniform probabilities for 2001 grid vocabulary
- Eval system used only single word class for non-spelled grid words (unlike IBM, CU).
- X/Y labeling of grid words gives additional 0.5% win over SRI2 (27.2% final WER).
Automatic Grid Word Tagging

- Problem: grid words are ambiguous
  - We are at bad and need, bad and need, versus
  - That's why we missed so bad
- Solution:
  - Build HMM tagger for grid words
  - Ambiguous grid words are generated by two states: GRIDLABEL or self.
  - State transitions given by trigram LM.
  - HMM parameters estimated from unambiguous words.
Other LM Issues

- Interpolating SPINE1 + SPINE2 models with optimized weighting is better than pooling data.
- Automatic grid word tagging is better than blindly replacing grid words with classes ("naïve" classes)
- Dry run performance, first decoding pass:

<table>
<thead>
<tr>
<th>Model/Data</th>
<th>Type</th>
<th>Perplexity</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMU trigram</td>
<td>Word</td>
<td>58.6</td>
<td>36.9</td>
</tr>
<tr>
<td>SRI trigram</td>
<td>Word</td>
<td>56.9</td>
<td>-</td>
</tr>
<tr>
<td>SPINE1+SPINE2</td>
<td>Word, interpolated</td>
<td>50.9</td>
<td>-</td>
</tr>
<tr>
<td>SPINE1+SPINE2</td>
<td>Class, naïve</td>
<td>43.7</td>
<td>31.7</td>
</tr>
<tr>
<td>SPINE1+SPINE2</td>
<td>Class, HMM-tagged</td>
<td>39.7</td>
<td>31.2</td>
</tr>
</tbody>
</table>
Word Posterior-based Decoding

- Word posterior computation:
  - N-best hypotheses obtained for each acoustic model
  - Hypothesis rescoring with new knowledge sources: pronunciation probabilities and class 4-gram LM
  - Hypotheses aligned into word confusion "sausages".
  - Score weights and posterior scaling factors jointly optimized for each system, for minimum WER

- Decoding from sausages:
  - Pick highest posterior word at each position
  - Reject words with posteriors below threshold (likely incorrect word, noise or background speech)
Word Posterior-based Adaptation and System Combination

- System combination:
  - Two or more systems combined by aligning multiple N-best lists into a single sausage (N-best ROVER)
  - Word posteriors are weighted averages over all systems
  - Final combination weights all three systems equally

- Adaptation:
  - 2 out of 3 systems were combined round-robin to generate improved hypotheses for model readaptation of the third system
  - Maintains system diversity for next combination step
Processing Steps

1. Segment waveforms.

2. Compute VTL and cepstral mean and variance normalizations.

3. Recognize using GI non-CW acoustic models and 3-gram multiword language models.

*Following steps are done for all 3 features*

4. Compute feature transformations for all speakers.

5. Recognize using transformed features.
Processing Steps

6. Adapt the CW and non-CW acoustic models for each speaker.

7. Use the non-CW acoustic models and 2-gram language models to generate lattices. Expand the lattices using 3-gram language models.

8. Dump N-best hypotheses from the lattices using CW speaker-adapted acoustic models.

9. Rescore the N-best using multiple KSs and combine them using ROVER to produce 1-best.
Processing Steps

10. Readapt the acoustic models using hypotheses from Step 9. For each feature model, use the hypotheses from the other two feature models.


12. Combine the N-best using N-best ROVER.
Processing Steps

Following steps are for SRI1 only

13. Adapt acoustic models trained on all data, including dry run data using the hypotheses from Step 12.


15. Combine all systems to generate final hypotheses. Do forced alignment to generate CTM file.
Results
### SPINE 2001 Dry Run Results

<table>
<thead>
<tr>
<th>Step</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3. Recognition with Mel features and non-CW GI models with 3-gram lm</td>
<td>31.6</td>
</tr>
<tr>
<td>Step 5. Recognition with transformed features and non-CW GI models with 3-gram lm</td>
<td>28.8 (Fourier)</td>
</tr>
<tr>
<td>Step 7. Generate lattices using speaker adapted non-CW models</td>
<td>24.9 (Fourier)</td>
</tr>
<tr>
<td>Step 8. Dump N-best from lattices using CW models</td>
<td>22.7 (Fourier)</td>
</tr>
<tr>
<td>Step 9. System Combination 1</td>
<td>19.5</td>
</tr>
<tr>
<td>Step 12. System Combination 2</td>
<td><strong>19.3</strong></td>
</tr>
</tbody>
</table>
## SPINE2001 Evaluation Results

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>SRI1/SRI2 (SRI lm1)</th>
<th>WER SRI3 (SRI lm2)</th>
<th>SRI4 (CMU lm, bug fixed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3.</td>
<td>Recognition with Mel features and GI models</td>
<td>39.0</td>
<td>38.6</td>
<td>42.8</td>
</tr>
<tr>
<td>Step 5.</td>
<td>Recognition with transformed features and GI models</td>
<td>Fourier 36.1</td>
<td>36.4</td>
<td>40.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mel 34.9</td>
<td>35.4</td>
<td>38.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLP 34.3</td>
<td>34.5</td>
<td>37.9</td>
</tr>
<tr>
<td>Step 8.</td>
<td>Dump N-best from lattices using CW models</td>
<td>Fourier 31.7</td>
<td>31.9</td>
<td>34.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mel 32.1</td>
<td>32.5</td>
<td>34.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLP 31.1</td>
<td>31.5</td>
<td>33.3</td>
</tr>
<tr>
<td>Step 9.</td>
<td>System Combination 1</td>
<td>28.0</td>
<td>28.1</td>
<td>30.0</td>
</tr>
<tr>
<td>Step 12.</td>
<td>System Combination 2</td>
<td>27.7 (SRI2)</td>
<td>28.0</td>
<td></td>
</tr>
<tr>
<td>Step 15.</td>
<td>System Combination 3</td>
<td>27.6 (SRI1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What Worked?

- Improved segmentation:
  - New segments were less than 1% absolute worse in recognition than true (reference) segments.
  - Last year, we lost 5.4% in segmentation.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Energy based (Eval2000)</th>
<th>Foreground/background recognizer</th>
<th>FG/BG recognition +reject word removal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eval 2000</td>
<td>31.5</td>
<td>36.9</td>
<td>34.2</td>
</tr>
<tr>
<td>Dry Run 2001</td>
<td>31.3</td>
<td>37.5</td>
<td>33.6</td>
</tr>
<tr>
<td>Eval 2001</td>
<td>38.2</td>
<td>-</td>
<td>39.5</td>
</tr>
</tbody>
</table>
What Worked? (continued)

- Feature SAT
  - Typical win was 4% absolute or more.

- 3-way system combination.
  - WER reduced by 3% absolute or more.

- Class-based language model
  - Improvement of 2%, 4-5% in early decoding stages.

- Acoustic model parameter optimization
  - Win of 2% absolute or more.
What Worked? (continued)

- MMIE training
  - MMIE trained acoustic models were about 1% abs. better than ML trained models.

- Word rejection with posterior threshold
  - 0.5% win in segmentation
  - 0.1% win in final system combination

- Acoustic readaptation after system combination
  - 0.4% absolute win.

- SPINE2001 system was about 15% absolute better than our SPINE2000 system.
SPINE1 Performance

- SPINE1 evaluation result: 46.1%
- SPINE1 workshop result: 33.7%
  - Energy-based segmentation
  - Cross-word acoustic models
- Current system on SPINE1 eval set: 18.5%
  - Using only SPINE1 training data
What Did Not Work

- Spectral subtraction
- Duration modeling
  - Marginal improvement, unlike our Hub5 results
  - Too little training data?
- Dialog modeling
  - Small win observed in initial experiments but no improvement in dry run.
Fourier Cepstrum Revisited

- Fourier cepstrum = IDFT(Log(Spectral Energy))
- Past research (Davis & Mermelstein 1980) showed that Fourier cepstrum is inferior to MFC.
- None of current ASR systems use Fourier cepstra.
- Our experiments support this, but we also found that adaptation can improve the performance significantly.


## Fourier cepstral features (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Dry Run 2001 WER</th>
<th>Eval 2001 WER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fourier</td>
<td>Mel</td>
</tr>
<tr>
<td>Step 3. Recognition with non-CW GI models and 3-gram lm</td>
<td>36.6</td>
<td>31.3</td>
</tr>
<tr>
<td>Step 5. Recognition with transformed features and non-CW SAT GI models with 3-gram lm</td>
<td>28.8</td>
<td>27.1</td>
</tr>
<tr>
<td>Step 7. Generate lattices using speaker adapted non-CW models</td>
<td>24.9</td>
<td>24.5</td>
</tr>
<tr>
<td>Step 8. Dump N-best from lattices using CW models</td>
<td>22.7</td>
<td>23.5</td>
</tr>
</tbody>
</table>
Fourier cepstral features (continued)

- Why does feature adaptation produce significant performance improvement?
  - Does DCT decorrelate features better than DFT?
  - What is the role of frequency warping in MFC?
- Can we reject any new feature based on a single recognition experiment?
Evaluation Issues

- System development was complicated by lack of proper development set (that is not part of the training set).

- Suggestion: use previous year's eval set for development (assuming task stays the same).

- Make standard segmenter available to sites who want to focus on recognition.
Future Work

- Noise modeling
- Optimize front-ends and system combination for noise conditions
- New features
- Language model is very important, but task-specific: how to "discover" structure in the data?
- Model interaction between conversants
Conclusions

- 15% abs. improvement since SPINE1 Workshop.
- Biggest winners:
  - Segmentation
  - Acoustic adaptation
  - System combination
  - Class-based language modeling
- Contrary to popular belief, Fourier cepstrum performs as well as MFCC or PLP.
- New features need to be tested in a full system!