Optimisation of Fast LVCSR Systems

Gunnar Evermann, Phil Woodland &
Rest of the CU-HTK STT team

December 5th 2003

Cambridge University Engineering Department
Overview

• Introduction

• 2003 CU-HTK 10xRT CTS system: structure, results and analysis

• Speed/accuracy trade-off

• Tuning lattice size

• System Combination & Pruning rescoring branches

• Conclusions
Introduction

• Current CU-HTK CTS “fast” system runs at 10xRT and based on models from full (200xRT) system

• Performance is about 5-7% relative worse than full system

• Target in 4 years is 1xRT while sustaining rate of accuracy improvements

• Achieving target relies on
  – much faster computers
  – better acoustic models (fancy techniques, more data)
  – more acoustic models for system combination
  – better LMs (higher-level knowledge, more data)
  – optimised software (decoders, adaptation, etc.)
  – improved system structure (can’t run dozens of systems and cross-adapt)
General system structure for 10xRT (BN/CTS)

- Segmentation
- Initial transcription 1xRT
- Normalisation (re-segment, VTLN, etc.) Adaptation 0.5xRT
- Lattice generation with word fourgram LM 4xRT
- Lattice rescoring: for each model set: 2xRT
  - Adaptation: MLLR (1-best + lattice), FV
  - Lattice rescoring
  - Confusion network generation
- System combination
Choosing Rescoring Model Sets

- Select 2 models from Four MPE triphone sets
  
  **A:** SAT HLDA  **B:** HLDA  **C:** SPron HLDA  **D:** non-HLDA

Results of pairwise system combination using CNC:

<table>
<thead>
<tr>
<th>System</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>23.0</td>
<td>23.6</td>
<td>23.4</td>
<td>24.8</td>
</tr>
<tr>
<td>+A</td>
<td>23.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+B</td>
<td></td>
<td>22.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+C</td>
<td></td>
<td></td>
<td>22.8</td>
<td></td>
</tr>
</tbody>
</table>

Individual Systems and pairwise combination

%WER on cts-eval02 after lattice-MLLR/FV and CN

- Best 3-way combination (A+C+D) gave 22.4
Error Analysis: Variation in Speaker WER

- The speaker WER varies widely, SAT and SPron WER are highly correlated but there are outliers

SAT and SPron %WER on cts-eval02
P1 (initial transcription): Speed/accuracy trade-off

• Accuracy of initial pass has little influence on overall result

<table>
<thead>
<tr>
<th>P1 speed xRT</th>
<th>P1</th>
<th>P2 trigram</th>
<th>P2 fourgram</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.48</td>
<td>37.4</td>
<td>26.3</td>
<td>25.5</td>
</tr>
<tr>
<td>0.83</td>
<td>35.2</td>
<td>26.3</td>
<td>25.4</td>
</tr>
<tr>
<td>1.50</td>
<td>34.4</td>
<td>26.1</td>
<td>25.2</td>
</tr>
</tbody>
</table>

P1 speed-accuracy trade-off (CTS eval02)

• In eval chose middle operating point for safety

⇒ Should have used fast setup and use time elsewhere
**P2 (lattice generation): Tuning lattice size**

use “Oracle” to find path with lowest WER (compared to reference) in lattice

![Graph showing Oracle word error rate against lattice density](image)

- Larger rescoring lattices are more likely to contain the correct answer...
Tuning lattice size (cont’d)

• …but we probably won’t find it anyway:

Oracle Search WER: rescore big lattices and take result as “reference” for oracle

Lattice search word error rate against lattice density (CTS eval02, P2-fg)
P3 (lattice rescoring): Predicting rescoring time

- To hit xRT target it is useful to predict rescoring time (P3) and prune lattices accordingly

![Graph showing rescoring runtime against lattice density & fit of log function (CTS eval02)]

Rescoring runtime against lattice density & fit of log function (CTS eval02)

- Curves are roughly log-shaped

- Reason: size of search network grows logarithmically with lattice size
System Combination

- Overall system combination helps, but not on all segments
- In the 2003 system 2-way combination SAT+SPRON
- Order of processing: latgen, SAT, SPron, combination
- SAT and SPron 1-best often identical
  \[\Rightarrow\] no gain from CNC
- example eval02: 6388 segments
- 1-best identical in 3824 segments (60%)
Pruning Rescoring Branches

- even if 1-bests differ often CNC output same as SAT hypothesis

- take final CNC output as reference and compare with earlier passes

<table>
<thead>
<tr>
<th></th>
<th>Word Accuracy</th>
<th>Sent Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2 trigram</td>
<td>88.8</td>
<td>57.5</td>
</tr>
<tr>
<td>P2 4-gram</td>
<td>90.1</td>
<td>60.1</td>
</tr>
<tr>
<td>P3.1 SAT</td>
<td>94.9</td>
<td>71.9</td>
</tr>
<tr>
<td>P3.2 SPron</td>
<td>95.2</td>
<td>71.9</td>
</tr>
</tbody>
</table>

- idea: try to predict for which segments CNC output is same as SAT hypothesis. prune further rescoring branches for these segments.

- train decision tree to predict that SAT and CNC 1-best are the same
Pruning Rescoring Branches (cont’d)

- information available: system output up to P3.1 (i.e., P1, P2, P3.1)
- features: length, confidence scores, \#words change in hypotheses
- best predictors: minimum confidence score and similarity of SAT and P2 hyps
- trained tree on eval02 & choose thresholds (skip 64% of segments)
- test on eval03: skip 66% segments, 43% audio, 32% rescoring runtime
  i.e. segments are short and easy.
  \[ \Rightarrow < 0.1\% \text{ WER change} \]
New 10xRT system

Changes:

- Faster P1 configuration
- Use SPron model for lattice generation (about 10% faster)
- Interpolate word 4-gram with class trigram
- Adaptively prune rescoring branches
- Add third branch: non-HLDA MPE MPron

ongoing, current results:

- P2 SPron is 0.3% better and faster
- SAT, SPron and 2-way combination 0.1% better
Future Work

• Prune branches more aggressively

• Choose rescoring models for each speaker

• Optimise models (HMMs and LMs) for fast systems