Speech Recognition
Lecture 10: Search.

Mehryar Mohri
Courant Institute of Mathematical Sciences
mohri@cims.nyu.edu
Speech Recognition Components

- **Acoustic and pronunciation model:**

  \[
  \Pr(o \mid w) = \sum_{d,c,p} \Pr(o \mid d) \Pr(d \mid c) \Pr(c \mid p) \Pr(p \mid w).
  \]

  - \(\Pr(o \mid d)\): observation seq. \(\leftarrow\) distribution seq.
  - \(\Pr(d \mid c)\): distribution seq. \(\leftarrow\) CD phone seq.
  - \(\Pr(c \mid p)\): CD phone seq. \(\leftarrow\) phoneme seq.
  - \(\Pr(p \mid w)\): phoneme seq. \(\leftarrow\) word seq.

- **Language model:** \(\Pr(w)\), distribution over word seq.
Continuous Speech Models

Graph topology: 3-state HMM model: for each CD phone $a e_b, d$.

Interpretation: beginning, middle, and end of CD phone.

Continuous case: transition weights based on distributions over feature vectors in $\mathbb{R}^N$, typically with $N = 39$. 

(Rabiner and Juang, 1993)
HMM model - Representation

- **Composite model**: obtained by taking the union and closure of all CD phone models.

\[ \left( \sum_{p=1}^{P} H_i \right)^* \]

- **Illustration:**

Tying can reduce the size.
CD Model Representation

(Deterministic transducer representation)

(MM, Pereira, Riley, 2007)
Pronunciation Dictionary

- **Phonemic transcription**
  - **Example:** word *data* in American English.
    - data  D ey dx ax  .32
    - data  D ey t ax  .08
    - data  D ae dx ax  .48
    - data  D ae t ax  .12

- **Representation**

```plaintext
  0  d:ε/1.0  1  ey:ε/0.4  ae:ε/0.6  2  dx:ε/0.8  t:ε/0.2  3  ax:data/1.0  4/1
```
N-Gram Models - Representation

Making a backoff model from counts

• Program:
grmmake foo.2g.counts.fsm > foo.2g.lm.fsm

• Graphical Representation:

3

bye/0.698

1

bye/0.698

bye/1.098

ε/3.500

4

bye/1.108

</s>/0.410

ε/4.481

hello/0.698

</s>/0.810

2

ε/4.704

hello/1.504

</s>/0.005

0/0
Recognition Cascade

- Combination of components

\[
\hat{w} = \arg\max_w \sum_{d,c,p} \Pr[o \mid d] \Pr[d \mid c] \Pr[c \mid p] \Pr[p \mid w] \Pr[w]
\]

\[
\approx \arg\max_w \max_{d,c,p} \Pr[o \mid d] \Pr[d \mid c] \Pr[c \mid p] \Pr[p \mid w] \Pr[w].
\]
Model Combination

Steps:

- models represented by weighted transducers.
- Viterbi approximation: semiring change.
- composition of weighted transducers.

\[
w = \arg\min_w \Pi_2 \left[ O \ast (H \circ C \circ L \circ G) \right].
\]
Search Problem

Problem:

• size of composed transducer prohibitively large.
• visiting all states and transitions impractical.
• how to combine models efficiently and return the best transcription?

Consequences:

• pruning.
• search errors.
Viterbi Algorithm

Specific shortest-distance algorithm

\[
d[q, t] = \min_{e: n[e] = q} [d[p[e], t] + w[e]]
\]
Beam Pruning

- **Time-synchronous beam search**: at each time $t$ keep only states within a fixed threshold $\beta$ of the best.

\[
d[q, t] \leq \min_{q \text{ explored}} d[q, t] + \beta
\]
Search Modes

- **On-the-fly composition**: $H \circ C \circ L \circ G$.
  - **Advantages**: Components can be modified, e.g., dynamic grammars. Memory usage.

- **Off-line composition**: full $H \circ C \circ L \circ G$ or parts.
  - **Advantage**: Recognition transducer optimization.
Key Optimization Ideas

- **General algorithms**: as opposed to *ad hoc* solutions.
  - Recognition transducer redundancy: use determinization to reduce or eliminate redundancy. **But**: not all weighted transducers are determinizable.
  - Recognition transducer size: use minimization to reduce space.
  - Recognition transducer weight distribution: use weight pushing to standardize weight distribution.
Disambiguation & Determinizability

- Determinizability of $L \circ G$: use auxiliary symbols to deal with homophones and unbounded delay. Transformation $L \rightarrow \tilde{L}$ according to:

  \[
  \begin{array}{ccc}
  r & e & h & d & \#_0 & read \\
  r & e & h & d & \#_1 & red
  \end{array}
  \]

- Determinizability of $C \circ L \circ G$: self-loops used to propagate auxiliary symbols to context-dependency level, $C \rightarrow \tilde{C}$.

- Determinizability of $H \circ C \circ L \circ G$: self-loops at initial state, auxiliary CD symbols mapped to new distinct distribution names, $H \rightarrow \tilde{H}$.
Recognition Transducer Optimization

(MM and Riley, 2001; MM, Pereira and Riley, 2007)

- Optimization cascade:

\[ N = \text{push}(\sigma_\epsilon(\min(\det(\tilde{H} \circ \det(\tilde{C} \circ \det(\tilde{L} \circ G))))))). \]

replace auxiliary symbols by \( \epsilon \)

- Other more general methods for making weighted transducers determinizable (Allauzen and MM, 2004).
Example - G

0 — jim/1.386 — jill/0.693 — bill/1.386 — 1 — read/0.400 — wrote/1.832 — fled/1.771 — 2/0
Example - \( L \)
Example

\[ \tilde{L} \circ G \]

\[ \det(\tilde{L} \circ G) \]

\[ \min(\det(\tilde{L} \circ G)) \]

\[ \text{push}(\sigma_\epsilon(\min(\det(\tilde{L} \circ G)))) \]
Recognition Transducer Standardization

- Minimal deterministic weighted transducers: unique up to state renumbering and to any weight and output label redistribution that preserves the total path weights and output strings.

- Weight-pushed transducer: selects a specific weight distribution along paths while preserving total path weights.

- Result is a standardized recognition transducer.
Factoring

Idea:

• decoder feature: separate representation for variable-length HMMs (time and space efficiency).
• To take advantage of this feature, factor integrated transducer $N = H' \circ F$.

Algorithm:

• Replace input of each linear path in $N$ by a single label naming an $n$-state HMM.
• Define gain of the replacement of linear path:

$$G(\sigma) = \sum_{\pi \in \text{Lin}(N), i[\pi] = \sigma} |\sigma| - |o[\pi]| - 1.$$
## 1st-Pass Recognition Networks
### 40K NAB Task

<table>
<thead>
<tr>
<th>network</th>
<th>states</th>
<th>transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>1,339,664</td>
<td>3,926,010</td>
</tr>
<tr>
<td>$L \circ G$</td>
<td>8,606,729</td>
<td>11,406,721</td>
</tr>
<tr>
<td>$\text{det}(L \circ G)$</td>
<td>7,082,404</td>
<td>9,836,629</td>
</tr>
<tr>
<td>$C \circ \text{det}(L \circ G)$</td>
<td>7,273,035</td>
<td>10,201,269</td>
</tr>
<tr>
<td>$\text{det}(H \circ C \circ L \circ G)$</td>
<td>18,317,359</td>
<td>21,237,992</td>
</tr>
<tr>
<td>$F$</td>
<td>3,188,274</td>
<td>6,108,907</td>
</tr>
<tr>
<td>$\text{min}(F)$</td>
<td>2,616,948</td>
<td>5,497,952</td>
</tr>
</tbody>
</table>
## 1st-Pass Recognition Networks

### 40K NAB Eval '95

<table>
<thead>
<tr>
<th>transducer</th>
<th>x real-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C \circ L \circ G$</td>
<td>12.5</td>
</tr>
<tr>
<td>$C \circ \det(L \circ G)$</td>
<td>1.2</td>
</tr>
<tr>
<td>$\det(H \circ C \circ L \circ G)$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\text{push}(\text{min}(F))$</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Recognition speed of the first-pass networks in the NAB 40,000-word vocabulary task at 83% word accuracy.
Rescoring

cheap model → lattice

n best → detailed model

rescoring
## 2nd-Pass Recognition Speed

### 160K NAB Eval '95

<table>
<thead>
<tr>
<th>network</th>
<th>x real-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C \circ L \circ G$</td>
<td>.18</td>
</tr>
<tr>
<td>$C \circ \text{det}(L \circ G)$</td>
<td>.13</td>
</tr>
<tr>
<td>$C \circ \text{push}(\min(\text{det}(L \circ G)))$</td>
<td>.02</td>
</tr>
</tbody>
</table>

Recognition speed of the second-pass networks in the NAB 160,000-word vocabulary task at 88%.
Effect of Vocabulary Size
NAB Eval '95

Bigram recognition results for vocabularies of (1) 10,000 words, (2) 20,000 words, (3) 40,000 words, and (4) 160,000 words. (LG Optimized Only.)
Effect of N-gram Order
NAB Eval '95

Recognition results for a (1) 10,000 word bigram, (2) 10,000 word trigram, (3) 40,000 word bigram, and (4) 40,000 word trigram. (LG Optimized Only.)
Effect of Shrink Parameter
NAB Eval '95

Recognition results for shrink factors (Seymore & Rosenfeld, 1996) of 5, 10, and 40.
Effect of Pushing 1st Pass, 40K NAB Eval '95

40K-word NAB 1st-pass recognition, 6,108,907-transition determinized and factored HMM-to-word transducer [Alpha 21284].
Effect of Pushing
2nd-Pass, 160K NAB Eval '95

160,000-word vocabulary NAB task, weight-pushing determinized HMM-to-word transducer lattices [Alpha 21284].
100K Names Recognition

100K names recognition, the effect of weight-pushing [SGI~Origin~2000].
## Model Combination by Lattice Intersection - SWBD Eval '00

<table>
<thead>
<tr>
<th>Model/pass</th>
<th>Mod1</th>
<th>Mod2</th>
<th>Mod3</th>
<th>Mod4</th>
<th>Mod5</th>
<th>Mod6</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLLR</td>
<td>30.3</td>
<td>30.2</td>
<td>30.8</td>
<td>30.7</td>
<td>31.4</td>
<td>32.6</td>
</tr>
<tr>
<td>Combined</td>
<td>30.3</td>
<td>29.6</td>
<td>28.9</td>
<td>28.8</td>
<td>28.7</td>
<td>28.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model/pass</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Mod1</strong>: 30.3</td>
</tr>
<tr>
<td><strong>Mod2</strong>: 30.2</td>
</tr>
<tr>
<td><strong>Mod3</strong>: 30.8</td>
</tr>
<tr>
<td><strong>Mod4</strong>: 30.7</td>
</tr>
<tr>
<td><strong>Mod5</strong>: 31.4</td>
</tr>
<tr>
<td><strong>Mod6</strong>: 32.6</td>
</tr>
</tbody>
</table>
N-Best Sequences

- **Motivation**: rescoring.
  - first pass using a simple acoustic and grammar lattice or $n$-best list.
  - re-evaluate alternatives with a more sophisticated model or use new information.

- **General problem**:
  - speech recognition, handwriting recognition.
  - information extraction, image processing.
N-Shortest-Paths Problem

Problem: given a weighted directed graph $G$, a source state $s$ and a set of destination or final states $F$, find the $N$ shortest paths in $G$ from $s$ to $F$.

Algorithms:

- (Dreyfus, 1969): $O(|E| + N \log(|E|/|Q|))$.


- (Eppstein, 2002): $O(|E| + |Q| \log |Q| + N)$.

+ explicit representation of $N$ best paths: $O(|Q| N^2)$.
N-Shortest Strings ≠ N-Shortest-Paths

Problem: given a weighted directed graph $G$, a source state $s$ and a set of destination or final states $F$, find the $N$ shortest strings in $G$ from $s$ to $F$.

Example: NAB Eval 95.

<table>
<thead>
<tr>
<th>Thresh</th>
<th>Non-Unique</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>2.0</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>2.5</td>
<td>54</td>
<td>4</td>
</tr>
<tr>
<td>3.0</td>
<td>1536</td>
<td>48</td>
</tr>
</tbody>
</table>
References


References

