Speech and Audio Processing & Recognition

Fadi Biady

Presenting:

Weighted Final State Transducers in Speech Recognition
(Mohri et al, 2002)
Introduction

• Large-vocabulary speech recognition is based:
  – HMMs models
  – Pronunciation dictionaries
  – n-gram language models
  – ...

• Can be represented by weighted finite-state transducers

• Apply operations on them efficiently
• Weighted finite-state acceptor examples:

WFSA $A = (\Sigma, Q, E, i, F, \lambda, \rho)$
• Weighted finite-state transducer examples:

\[
WFST \ T = (\Sigma, \Omega, Q, E, i, F, \lambda, \rho)
\]
The Authors’ Approach for ASR

• Use a **uniform** transducer representation for
  – N-gram grammar
  – Pronunciation dictionary
  – Context-dependency specifications
  – HMM topology
  – Word
  – Phone
  – Lattices and n-best output list

• As opposed to a pre-programmed and pre-specified number of levels
Basic operations

1. Union
2. Concatenation
3. Kleene closure
4. Convert transducers to acceptors
5. Find the best/n-best in a weighted transducer
6. Remove unreachable states and transitions
7. Sort topologically
Weighted Transducer Algorithms - Composition

$R \circ S$

(a) 

(b) 

(0, 0) 

(1, 1) 

(1, 2) 

(3, 2)/1.3
Weighted Transducer Algorithms - Determinization
Weighted Transducer Algorithms - Determinization

• Why deterministic?

• Determinization and minimization give an efficient and compact representation of lexicon

• Not all weighted automata can be determinized

• 2 weighted acceptors are equivalent if they associate the same weight to each input string

• 2 WT are equivalent if the output is also the same.
Determinization

0 → 1 → 3
a/0 → b/1 → c/4 → a/3 → b/4 → c/7 → d/0 → e/1 → f/11 → f/13 → e/10

0 → 1 → 3
a/0 → b/1 → c/4 → d/0 → e/10 → e/11 → f/11
Determinization Algorithm

S={ (0,0) } 

a/min{0+0,0+3} 

e/min{0+0+2,3+10}

S={ (1,0-0),(2,3-0) }

=>

S={ (0,0) }

....
Minimization

(Aho et al., 1974)

Weight pushing

(Aho et al., 1974)
WFST Applications

- Transducer Combination
Context Dependent

- Triphone model
- Language model (e.g., 3-gram)
- Tripgram: each state \((a, b)\) encodes the past \((a)\) and the future \((b)\)
Transducer Combination for ASR

• For example:
  – H: Union of phone models (HMMs)
  – C: Context-dependency transducer from phones
  – L: pronunciation dictionary
  – G: word grammar

\[ H \circ C \circ L \circ G \]

• Transducer that maps distributions \( \rightarrow \) words restricted to \( G \).
Transducer Combination for ASR

• Determinization at each step of the composition of each pair of transducers
• For homophones add #i auxiliary phone
  – r eh d #0
  – r eh d #1
Experimental results

- **H**: Acoustic model of 7,208 distinct HMM states
- **C**: Triphonic context-dependency transducer with 1,525 states and 80,225 transitions.
- **L**: 40,000-word pronunciation dictionary
- **G**: Trigram language model with 3,926,010 transitions
<table>
<thead>
<tr>
<th>transducer</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>

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

- Accuracy 83% WER
- When using 2 passes: 88% WER
Thank you!