MTRG presentation (5 12 Mar 2008)

Chunk-level reordering of source lang. sentences with automatically learned rules for statistical machine translation

Yuqi Zhang
Richard Zens
Hermann Ney

NAACL/HLT 2007, SSST/AMTA workshop

presented by Jeremy G. Kahn
Syntax in MT [for ordering]

- **Source pre-ordering**
  - learned tree-tree reorderings [Xia & McCord '04]
  - hand-written source-tree rewrites [Collins et al. '05]
  - “Pre-translate” on P OS tags: [Costa-jussà & Fonollosa '06]

- **In decoder**
  - source coherence [Quirk et al.]
  - target tree structure [Knight et al.] and others

- **Reranking**
  - Syntactic bi-tree ordering feats indicate good cands [Chen et al. '06, Crego & Mariño '06]
Have cake, will eat too

- Chunk source
  - Shallow syntactic parse (no hierarchy)
  - Tag sequence = \{POS tag|chunk tag\}+

- Reorder source
  - Learn rules against chunk-tag sequence
  - But don't make a hard decision: reorder into source lattice, allowing non-reordered input as well
  - Add additional LM score \( p(S') \)
System diagram

Standard Translation Process

source text sentences

SMT system

translation output

Translation Process with Source Reordering

source text sentences

POS tagging

shallow chunking

reordering rules

source reordering lattice

SMT system

translation output
Chunker details

- Using ICTCLAS POS tagger
- Train YASMET on CTB chunks (first non-unary branch)
- 24 types of chunks, trained on 106K chunk exemplars (487K words)
- 74.5% per-word accuracy, 63.3% per-chunk F
Reordering rule extraction

• GIZA++ intersective alignments
• Merge into source-side chunks
• “Phrase” extraction, discarding cross-phrases
• All other chunk-to-word phrases are rule [templates], with monotonizing.
Reordering rules

<table>
<thead>
<tr>
<th>NP</th>
<th>Shanghai</th>
<th>NP</th>
<th>Development</th>
<th>with</th>
<th>Legislation</th>
<th>建设</th>
<th>并存</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>f0</td>
<td>f1</td>
<td>f2</td>
<td>f3</td>
<td>f4</td>
<td>f5</td>
<td>f6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td>#</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td>#</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>v</td>
<td>#</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>v</td>
<td>#</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td>v</td>
<td># 0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td>v</td>
<td># 1</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td>v</td>
<td># 2</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sentence Permutations

0 1 2 3 4 5 6
2 3 4 5 0 1 6
0 1 2 3 4 5 6
0 1 6 2 3 4 5
0 1 2 3 4 5 6
2 3 4 5 6 0 1
8 0 1 2 3 4 5
Cross phrases
Decoding

- Decoder, LM, TM not changed
  - No retraining of TM!
- Usual log-lin combination:
  - Phrase & word TM, phrase count
  - Phrase-length & word-length feats, distortion model
  - Word TM, 6gm p(T) LM (target)
  - \( p(S') \) reordered-source probability
- \( p(S') \) is trained on monotonized source chunks
Experimental setup

• IWSLT{04,05,06} task: Basic Traveling Expression Corpus (BTEC)
  – {16,16,7} refs per utterance(!)
  – BLEU, NIST, WER, PER reporting
  – Note: chunker out-of-domain

• Trained LM, TM, p(S') model on same corpus:
  – 40k sent train
  – 489 sent dev
  – {500,506,500} sent test
Experimental results

- Baseline: non-monotone
- Source-reordering (Chunk+POS rules)

Table 5: Translation performance for the Chinese-English IWSLT task

<table>
<thead>
<tr>
<th></th>
<th>WER[%%]</th>
<th>PER[%%]</th>
<th>NIST</th>
<th>BLEU[%%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWSLT04</td>
<td>baseline</td>
<td>47.3</td>
<td>38.2</td>
<td>7.78</td>
</tr>
<tr>
<td></td>
<td>source reordering</td>
<td>46.3</td>
<td>37.2</td>
<td>7.70</td>
</tr>
<tr>
<td>IWSLT05</td>
<td>baseline</td>
<td>45.0</td>
<td>37.3</td>
<td>7.40</td>
</tr>
<tr>
<td></td>
<td>source reordering</td>
<td>44.6</td>
<td>36.8</td>
<td>7.51</td>
</tr>
<tr>
<td>IWSLT06</td>
<td>baseline</td>
<td>67.4</td>
<td>50.0</td>
<td>6.65</td>
</tr>
<tr>
<td></td>
<td>source reordering</td>
<td>65.6</td>
<td>50.4</td>
<td>6.46</td>
</tr>
<tr>
<td></td>
<td>source reordering+non-monotone decoder</td>
<td>66.5</td>
<td>50.3</td>
<td>6.52</td>
</tr>
</tbody>
</table>

Improvements on BLEU, WER, but hurts NIST (!)

Using non-monotone decoder and source reordering hurts
Experimental results (2)

• Using POS-only vs. POS+Chunk

Table 6: Translation performance of reordering methods on IWSLT 2004 test set

<table>
<thead>
<tr>
<th></th>
<th>WER [%]</th>
<th>PER [%]</th>
<th>NIST</th>
<th>BLEU [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>47.3</td>
<td>38.2</td>
<td>7.78</td>
<td>39.1</td>
</tr>
<tr>
<td>POS</td>
<td>46.9</td>
<td>37.5</td>
<td>7.38</td>
<td>39.7</td>
</tr>
<tr>
<td>Chunk</td>
<td>46.3</td>
<td>37.2</td>
<td>7.70</td>
<td>40.9</td>
</tr>
</tbody>
</table>

• (POS+)Chunk better on all measures
  - How much was re-trained?
Final win: speed [& size]

- On 2006 IWSLT test set, decoding time:
  - Baseline: 17.5 min
  - Source-reordering: 12.3 min
    (unclear whether cost of reordering is included)

- Size wins on IWSLT-2004:

<table>
<thead>
<tr>
<th></th>
<th>avg. density</th>
<th>used rules</th>
<th>translation time [min/sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>15.7</td>
<td>6868</td>
<td>7:08</td>
</tr>
<tr>
<td>Chunk</td>
<td>8.2</td>
<td>3685</td>
<td>3:47</td>
</tr>
</tbody>
</table>
Unclear areas

• $p(S')$ score is learned from “reordered text”—are all reorderings applied? (I think so!)

• Comparison vs. POS-only source reordering: is $p(S')$ model recomputed?

• Speed comparison: reordering cost?
Questions

- Poor syntactic chunking gives win – how key is it that syntax used at all?
- What are formal differences between source-lattice reordering and decoder-reordering?
  - Source vs. target re-ordering
  - Search constraints
Extensions

• Better chunkers

• Rather than $p(S')$ per path, lattice could include (trainable) weight for particular reordering rule

• What linguistics are actually useful?
  – vs. random chunking
  – vs. high-MI chunking
  – vs. better chunking