Lexicalized Inversion Transduction Grammar for Alignment

Hao Zhang and Daniel Gildea

Computer Science Department
University of Rochester
Alignment Problem

Finding translation pairs in bitext sentences:

I see them

Je les vois

- Inversion Transduction Grammar (Wu 1997) has achieved state of the art results against gold standard alignments (Zhang & Gildea 2004)

- ITG is tree-based, but entirely data-driven
Outline

- Lexicalize ITG with word pairs
- Prune the chart for bitext parsing using inside-outside estimate
- Compare with ITG on word alignment
ITG Example

them

see

I

Je les vois

S

A

C

B

C

I/Je

see/vois

them/les

Zhang and Gildea
### ITG Rules

<table>
<thead>
<tr>
<th>Structural Rules</th>
<th>Lexical Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S → A</strong></td>
<td>C → e_i/f_j</td>
</tr>
<tr>
<td>A → [AB]</td>
<td></td>
</tr>
<tr>
<td>A → [BB]</td>
<td>C → ε/f_j</td>
</tr>
<tr>
<td>A → [CB]</td>
<td>C → e_i/ε</td>
</tr>
<tr>
<td>A → [AC]</td>
<td></td>
</tr>
<tr>
<td>A → [BC]</td>
<td></td>
</tr>
<tr>
<td>A → [CC]</td>
<td></td>
</tr>
<tr>
<td>B → ⟨AA⟩</td>
<td></td>
</tr>
<tr>
<td>B → ⟨BA⟩</td>
<td></td>
</tr>
<tr>
<td>B → ⟨CA⟩</td>
<td></td>
</tr>
<tr>
<td>B → ⟨AC⟩</td>
<td></td>
</tr>
<tr>
<td>B → ⟨BC⟩</td>
<td></td>
</tr>
<tr>
<td>B → ⟨CC⟩</td>
<td></td>
</tr>
</tbody>
</table>
Lexicalizing ITG

- Introduce head generation rules to associate word pairs with nonterminals

  \[ X \rightarrow X(e/f) \]

- Incorporate head selection rules with binary rules

- Parent and head child have the same lexicalization:

  \[ X(e/f) \rightarrow [Y(e/f) Z] \]

  \[ X(e/f) \rightarrow [Y Z(e/f)] \]

  \[ X(e/f) \rightarrow \langle Y(e/f) Z \rangle \]

  \[ X(e/f) \rightarrow \langle Y Z(e/f) \rangle \]
LITG Example

Zhang and Gildea
## Categorization of Alignment Parameters

<table>
<thead>
<tr>
<th></th>
<th>ITG</th>
<th>LITG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lexical translation</strong></td>
<td>P(C → e_i/f_j)</td>
<td>P(X → X(e/f))</td>
</tr>
<tr>
<td></td>
<td>P(C → ε/f_j)</td>
<td>P(X → X(ε/f_j))</td>
</tr>
<tr>
<td></td>
<td>P(C → e_i/ε)</td>
<td>P(X → X(e_i/ε))</td>
</tr>
<tr>
<td><strong>re-ordering</strong></td>
<td>P(A → [AB])</td>
<td>P(A(e_i/f_j) → [A B(e_i/f_j)])</td>
</tr>
<tr>
<td></td>
<td>P(B → ⟨BA⟩)</td>
<td>P(B(e_i/f_j) → ⟨B(e_i/f_j) A⟩)</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Smoothing

\[
P([Y(e/f) \ Z] \mid X(e/f)) =
(1 - \lambda)P_{EM}([Y(e/f) \ Z] \mid X(e/f))
+ \lambda P([Y(*) \ Z] \mid X(*))
\]

where

\[
\lambda = 1/(1 + \text{Expected-Counts}(X(e/f)))
\]
LITG Parsing

A DP framework for EM training and Viterbi alignment selection
Complexity

Time

- ITG is $O(n^6) = O(n^{3.2})$
- Choosing head word from both languages introduces additional complexity of $O(n^2)$
- LITG is $O(n^8) = O(n^6 \cdot n^2)$

Space

- ITG is DP over $O(n^4)$ bitext cells
- LITG is DP over $O(n^6) = O(n^4 \cdot n^2)$ cells
Pruning: General Strategy

- Use “Figure of Merit” to estimate whether a given cell likely to be used in final parse.
- Only try to build a cell from smaller cells if its FOM looks promising.
- Our FOM is based on IBM Model 1.
- Apply beam ratio pruning on bitext cells with common source substring.
Inside-outside Tic-tac-toe Pattern

If a cell \((l, m, i, j)\) is used in final parse, all alignments must be from shaded “tac-tac-toe” area:

\[
\begin{array}{|c|c|}
\hline
m & i \\
\hline
l & j \\
\hline
\end{array}
\]

Figure of Merit\((l, m, i, j)\) = INS\((l, m, i, j)\) · OUTS\((l, m, i, j)\)
Inside-outside Tic-tac-toe Pattern

If a cell \((l,m,i,j)\) is used in final parse, all alignments must be from shaded “tac-tac-toe” area:

\[
\text{Figure of Merit}(l, m, i, j) = \text{INS}(l, m, i, j) \cdot \text{OUTS}(l, m, i, j)
\]
Computing Inside Estimate

\[
\text{INS}(l, m, j, j) \\
= \text{INS}(l, m - 1, j, j) \\
+ t(f_j | e_m)
\]

\[
\text{INS}(l, m, i, j) \\
= \text{INS}(l, m, i, j - 1) \\
\cdot \text{INS}(l, m, j, j)
\]
Pruning Summary

- A total of $n^4 \cdot n^2$ cells in the dynamic programming table (chart)
- Tic-tac-toe pruning to reduce $n^4$ to $\delta n^4$
- Top-k pruning to reduce the $n^2$ lexicalized hypotheses within each of the $n^4$ cells to $k$
- $\delta k \cdot n^4$ cells to be computed after applying both
Speedup of EM from Pruning for Unlexicalized ITG

AER was not changed after pruning.

Zhang and Gildea
### Experiments

<table>
<thead>
<tr>
<th></th>
<th>Training Data</th>
<th>Evaluation Data</th>
<th>Pruning Parameters</th>
<th>Initial Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>LITG-15</td>
<td>6984</td>
<td>20</td>
<td>-,-</td>
<td>uniform</td>
</tr>
<tr>
<td>LITG-25</td>
<td>18,773</td>
<td>47</td>
<td>$10^{-5}$, 25</td>
<td>ITG-25</td>
</tr>
</tbody>
</table>

$$
\text{AER} = 1 - \frac{2 \cdot |A \cap G|}{|A| + |G|}
$$

G is the set of gold standard word links.

A is the set of machine-output word links.
### Results on \( \leq 15 \) word long English-Chinese sentence pairs

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>Alignment Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Model 1</td>
<td>.59</td>
<td>.37</td>
<td>.54</td>
</tr>
<tr>
<td>IBM Model 4</td>
<td>.63</td>
<td>.43</td>
<td>.49</td>
</tr>
<tr>
<td>ITG</td>
<td>.62</td>
<td>.47</td>
<td>.46</td>
</tr>
<tr>
<td>Lexicalized ITG</td>
<td>.66</td>
<td>.50</td>
<td>.43</td>
</tr>
</tbody>
</table>
### Results on ≤ 25 word long English-Chinese sentence pairs

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>Alignment Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Model 1</td>
<td>.56</td>
<td>.42</td>
<td>.52</td>
</tr>
<tr>
<td>IBM Model 4</td>
<td>.67</td>
<td>.43</td>
<td>.47</td>
</tr>
<tr>
<td>ITG</td>
<td>.68</td>
<td>.52</td>
<td>.40</td>
</tr>
<tr>
<td>Lexicalized ITG</td>
<td>.69</td>
<td>.51</td>
<td>.41</td>
</tr>
</tbody>
</table>

Zhang and Gildea
Summary

- LITG is a version of ITG where inversions are sensitive to actual words.
- Strengthening the structural component of ITG helps aligning short sentences.
- Possibly “trapped” in local maxima during training after pruning lexicalized constituents for long sentences.
- Discovery: Tic-tac-toe pruning is an effective way to reduce the number of bitext cells for bilingual parsing.
Thanks

Rebecca Hwa
Kevin Knight
Daniel Marcu
Big-Picture Seminar @ URCS
NL Seminar @ ISI