Translation as Parsing with a Synchronous Tree Grammar

Libin Shen
BBN Technologies

Feb. 24, 2009
Overview

- Improving HierDec
  - Problems with a string-to-tree model
  - Extension to a tree-to-tree model

- Preliminary results

- Conclusions and future work
The HierDec System

- HierDec is BBN’s Hierarchical MT Decoder
  - From source-string to target-dependency-structure
  - Extended the string-to-string approach of Hiero (Chiang, 2005)

- Main Components
  - Rule extractor
    - Input: bi-lingual training data with GIZA alignment target parse trees
    - Output: string to dependency transfer rules, e.g.

  - Decoder
    - A chart parsing algorithm that produces a shared forest of target dependency structures
    - Using a target dependency LM on the fly
Problems of Using Target LMs Only

- An example from the last OntoNotes workshop
  - Source: 从 机场 送 父母 回来
     from airport send parents to come back
  - Ref: coming back from the airport after sending his parents
  - Agile: from the airport to send their parents to come back

- Traditional string LMs prefer locally fluent translation

- Dependency LMs prefer grammatical translation

- A string-to-tree model does not employ any source side structural constraint to avoid incorrect application of translation rules
Solution: Using Source Dependency LM

- To generate source dependency structures in decoding to exploit the parallel source dependency relations
  - Building aligned dependency trees on both sides in parallel
  - Source dependency LM to measure the source analysis

Ref: *coming back* from the airport after sending *his parents*

Agile: from the airport to send *their parents* to *come back*

Source dependency for Ref and Agile

Decode

Output text

coming back from the airport after sending his parents
HierDec in Dependency-to-Dependency Mode

• Rule extraction
  – Input:
    • Bi-lingual training data with GIZA alignment
    • Target parse trees
    • Source parse trees
  – Output:
    • Dependency-to-dependency transfer rules, e.g.

• Decoding
  – A chart parser that produces a shared forest of
    • Target dependency structures
    • Aligned source dependency structures
  – Using the source dep. LM score as an extra feature
Experiments on MT06 and MT08

- **Experimental setup:**
  - Test: MT06 and MT08 Chinese-English
  - Development: MT02-05, tuned on IBM BLEU
  - Training: GALE data with GIZA alignment; Source and target parse trees generated by two independent parsers.

- **Results:**
  - The dependency-to-dependency model does not show any improvement over the baseline, but they are very close.

<table>
<thead>
<tr>
<th>Model</th>
<th>MT06 BLEU</th>
<th>MT06 TER</th>
<th>MT08 BLEU</th>
<th>MT08 TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>string-to-dep</td>
<td>37.44</td>
<td>54.64</td>
<td>33.05</td>
<td>56.79</td>
</tr>
<tr>
<td>dep-to-dep</td>
<td>37.30</td>
<td>54.24</td>
<td>33.03</td>
<td>56.59</td>
</tr>
</tbody>
</table>
Experiments on MT06 and MT08 (cont’)

• Abnormal decrease in the number of transfer rules
  – The number is supposed go up: One string-to-dep rules would be splitted into several dep-to-dep rules due to the different analyses of the source dependency.
  – However, many more translation rules are discarded since the source side cannot be represented as a *well-formed* dependency structure.

• Cause of the phenomenon:
  – The source and target trees generated by two independent parsers are inconsistent.

• The missing translation rules may result in the performance degradation.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>string-to-dep</td>
<td>41,013,346</td>
</tr>
<tr>
<td>dep-to-dep</td>
<td>39,213,131</td>
</tr>
</tbody>
</table>
Conclusions

• Parsing with a synchronous tree grammar (Shieber and Schabes, 1990) is empirically a tractable decoding algorithm for statistical MT.
  – A neat solution to employ source and target dependency relations jointly.

• Our first attempt does not shown improvement over a state-of-the-art string-to-dependency model, but it is promising.
  – Fixing existing flaws in the parsing model will give rise to performance improvement.
Future Work in This Approach

• A bi-lingual parser
  – Trained from bi-lingual treebanks.
  – To parse the MT bi-lingual training data, and it guarantees better source and target tree consistency.
  – Self-training of the bi-lingual parser.

• To integrate alignment with bi-lingual parsing
  – From word-level alignment to hierarchical alignment