Preordering for Statistical Machine Translation

Automatically Learning Source-Side Reordering Rules

Riashat Islam

University of Cambridge

MLSALT8 Statistical Machine Translation, MPhil Machine Learning, Speech and Language Technology
Introduction

Automatically Learning Preordering Rules
  Introduction
  Approach
  Evaluation

Neural Preordering

Comparison of Results

Summary
Preordering in Machine Translation

- Reorder the source sentence such that it best resembles the order of the target sentence
- Preprocessing step in phrase-based machine translation
- Transform source-sentence into target-like order

Benefits:

- Better translation models
- Speeds up decoding
- Improved translation quality for language pairs
- Benefical for real-time commercial systems
We have come to quite like Xi’an.

私たちは、すっかり西安が好きになりました。
We have come to quite like Xi’an.

We quite Xi’an like to come have.

私たちは、すっかり西安が好きになりました.
Automatically Learning Source-Side Reordering Rules for Large Scale Machine Translation (Genzel, 2010)
Automatically Learning Preordering Rules

- Automatically learn the reordering rules
- Language-independent preordering using source-side language parsers
- Learn rules for 8 different language pairs
- Important word order transformations can be captured using this approach

**Goal:** To find a method that works for many language pairs regardless of word order transformations needed and without language specific tuning
Related Approaches to Preordering

1. Other preprocessing based reordering approaches
   - Manually written rules for different languages
   - Common language pair, German-English (Collins, 2005)
   - Automatically learning reordering patterns for French-English (Xia and McCord, 2004)
   - Learn reordering rules based on sequences of part-of-speech tags (Rottmann and Vogel, 2007)

2. Automatically aligning source-parsed data
   - Feature-rich logistic regression model (Jehl, 2014)
   - Neural Network to learn node swapping model (Gispert, 2015). Briefly later...
Approach

- Reordering of source side of training and test data
- Produce a parse tree on the source side
- Reordering of nodes in a parse tree of the source sentence
- Each reordering described by series of rules

Overview

- Automatically learn a rule series for each language pair
- Tree transformed sequentially
- Reordered sentence read off the leaves of the tree
Approach

Figure 1: Parse tree of a sentence and its reordering

(a) A sample parse tree
(b) After reordering (moving RB over _NN)
Rule Space and Evaluation Metric

Rule Space

- Two pairs of a rule - Conditioning Context and Action
- Node matched against conditioning context
- Each condition is a feature-value pair
- Action on the node for a found match - swap the children of the node

Evaluation Metric

- Evaluate quality of given reordering rule
- Cross alignment links for a given aligned sentence pair
- \textit{Estimated BLEU} gain
Algorithm

- Word-aligned sentence pairs as inputs
- Consider all rules
- Append the best new rule to the sequence according to a metric
- Consider several variations that produce a different sequence of rules
- Choose the variation that performs best on development set

Several optimization variants for appending rules in the sequence

- Optimizing Crossing Score
- Optimizing Estimated BLEU gain
- Optimizing Estimated BLEU gain in sequence
Algorithm 1 Optimizing alignment links

input: A set of aligned sentence pairs
base = <empty sequence>;
for several iterations do
  candidate_rules = GenerateAllCandidateRules(input, base);
  base.append(MinCost(candidate_rules))
end for
Train system from English to 7 other languages
Evaluate three variants of the algorithm
Each algorithm outputs a reordering rule sequence which is applied to all the training and test data
Proceed with phrase-based SMT, complete system trained from scratch - baseline decoder can also apply local reordering of upto 4 words
BLEU scores for each of the algorithms

Languages Used: SOV languages (Japanese, Korean, Hindi), VSO language (Welsh), Long Distance Verb Movement (German), Noun modifier issues (Russian and Czech)
### Evaluation and Results

**Figure:** Evaluation of Manual vs Automatic Reordering

<table>
<thead>
<tr>
<th>Language</th>
<th>Base</th>
<th>Manual</th>
<th>Automatic</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hindi</td>
<td>16.85</td>
<td>19.25</td>
<td>19.36</td>
<td>0.11</td>
</tr>
<tr>
<td>Japanese</td>
<td>25.91</td>
<td>28.78</td>
<td>29.12</td>
<td>0.34</td>
</tr>
<tr>
<td>Korean</td>
<td>23.61</td>
<td>27.99</td>
<td>27.91</td>
<td>-0.08</td>
</tr>
</tbody>
</table>
### Evaluation and Results

![Table showing BLEU scores for different languages across various systems.](image)

<table>
<thead>
<tr>
<th>Language</th>
<th>Google %BLEU</th>
<th>Base %BLEU</th>
<th>Var. 1 gain</th>
<th>Var. 2 gain</th>
<th>Var. 3 gain</th>
<th>Best on dev gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Czech</td>
<td>16.68</td>
<td>15.35</td>
<td>-0.08</td>
<td>0.13</td>
<td>0.19</td>
<td>0.19</td>
</tr>
<tr>
<td>German</td>
<td>20.34</td>
<td>18.65</td>
<td>0.47</td>
<td>0.30</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>Hindi</td>
<td>19.15</td>
<td>16.85</td>
<td>2.25</td>
<td>2.08</td>
<td>0.15</td>
<td>2.08</td>
</tr>
<tr>
<td>Japanese</td>
<td>30.74</td>
<td>25.91</td>
<td>3.05</td>
<td>2.60</td>
<td>3.05</td>
<td>3.05</td>
</tr>
<tr>
<td>Korean</td>
<td>27.99</td>
<td>23.61</td>
<td>3.34</td>
<td>3.77</td>
<td>4.16</td>
<td>4.16</td>
</tr>
<tr>
<td>Russian</td>
<td>16.80</td>
<td>15.33</td>
<td>0.08</td>
<td>0.10</td>
<td>0.10</td>
<td>0.08</td>
</tr>
<tr>
<td>Welsh</td>
<td>27.38</td>
<td>25.48</td>
<td>1.25</td>
<td>0.77</td>
<td>1.43</td>
<td>1.43</td>
</tr>
</tbody>
</table>

**Figure:** Comparison of Results on internal test set for 3 systems
Evaluation and Results

<table>
<thead>
<tr>
<th>Language</th>
<th>Decoder reordering</th>
<th>No decoder reordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Czech</td>
<td>0.21</td>
<td>0.08</td>
</tr>
<tr>
<td>German</td>
<td>0.72</td>
<td>0.55</td>
</tr>
<tr>
<td>Hindi</td>
<td>2.51</td>
<td>2.27</td>
</tr>
<tr>
<td>Japanese</td>
<td>3.21</td>
<td>3.21</td>
</tr>
<tr>
<td>Korean</td>
<td>4.30</td>
<td>4.15</td>
</tr>
<tr>
<td>Russian</td>
<td>0.14</td>
<td>-0.10</td>
</tr>
<tr>
<td>Welsh</td>
<td>1.34</td>
<td>0.98</td>
</tr>
</tbody>
</table>

**Figure:** Difference against baseline system in % BLEU gain
Using neural networks for source-side preordering (Gispert, 2015)
- Based on using a logistic regression model (Jehl, 2014)

Neural Networks to learn a node-swapping model
- Estimate node swapping probability in a parse tree using a logistic regression model
- Better modelling capabilities of NNs to achieve better performance at preordering
- Improves translation performance across various language pairs

First steps towards using NNs in preordering for SMT
### Comparing Experimental Results

- Baseline (No Preordering)
- Rule-based approach (Genzel, 2010)
- Linear model logistic regression (Jehl, 2014)
- Neural Network based preordering (Gispert, 2015)

<table>
<thead>
<tr>
<th>System</th>
<th>eng-jpn</th>
<th>eng-kor</th>
<th>eng-chi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>50.9</td>
<td>28.7</td>
<td>44.8</td>
</tr>
<tr>
<td>Genzel(2010)</td>
<td>54.0</td>
<td>30.5</td>
<td>45.4</td>
</tr>
<tr>
<td>Jehl(2014)</td>
<td>55.0</td>
<td>33.1</td>
<td>45.8</td>
</tr>
<tr>
<td>Gispert(2015)</td>
<td>55.6</td>
<td>33.4</td>
<td>46.5</td>
</tr>
</tbody>
</table>
(Genzel, 2010) used automatic preordering as a preprocessing step generalized for various language pairs.

Neural Network based preordering obtains best BLEU scores across all language pairs.

Preordering approaches are independent of the MT system used.

Preordering helps obtain faster decoding times.

Improvements in translation quality with better BLEU scores.
Thank You... Questions?