A Dependency-to-String Model for Chinese-Japanese SMT System

Hua Shan   Lu Bai   Te Luo   Yujie Zhang
School of Computer and Information Technology, Beijing Jiaotong University
No.3 Shangyuancun, Haidian District, Beijing, China, 100044
{13120422, 13120397, 14120472 , yjzhang} @bjtu.edu.cn

Introduction

- Our system employs a dependency-to-string translation method for Chinese-Japanese SMT.
  ➢ Motivated by that dependency grammar holds both syntactic and semantic knowledge.
- Our system achieves a BLEU of 34.87 and a RIBES of 79.25 on the Chinese-Japanese translation task in the official evaluation.
  ➢ Our dependency-to-string model improved the BLEU score by 0.62 and the RIBES score by 0.31 on the test set.
  ➢ The evaluation results illustrated that the translation system based on the dependency-to-string model is effective.

Decoding

Our decoder is based on bottom up chart parsing algorithm that convert the input dependency structure into a target string. It finds the best derivation among all possible derivations $D$. Given a source dependency structure $T$, the decoder traverses each internal node $n$ of $T$ in post-order. And we process it as follows.

1) If $n$ is leaf node, it checks the translation rules $H$ and uses the matched rules to generate candidate translation.
2) If $n$ is a internal node, it enumerates all instances of the clauses or phrases of the HDR fragment rooted at $n$, and checks the translation rules for matched translation rules. If there is no matched rules, we construct a pseudo translation rule according to the word order of the HDR fragment in the source side.
3) Make use of Cube Pruning algorithm to generate the candidate translation for the node $n$.

Experiment and Evaluation

We used Stanford Word Segmenter for Chinese word segmentation, with the standard of CTB, and used Stanford Parser for Chinese dependency parsing. We used JUMAN for Japanese word segmentation and used SRI Language Modeling Toolkit for training a 4-gram language model on the Japanese corpus preprocessed. We obtained the word alignments by running GIZA++ on the corpus in both directions and applying “grow-diag-and” refinement. We make use of MERT to tune the feature weights in order to maximize the system’s BLEU score on the development set.

Translation Model

For a given source language dependency tree $T$, the model may generate more than one derivations $D$ that convert a source dependency tree $T$ into a target string $e$, thus producing a large amount of candidate translations. We adopt a general log-linear model to evaluate the candidate translations as:

$$P(D) \approx \prod \phi(D) \cdot \prod \lambda_i$$

where $\phi_i$ are feature functions defined on derivation $D$ and $\lambda_i$ are the feature weights.

The Table shows the number of the extracted translation rules and the translation performance on the test data. Furthermore, we implemented a MOSES PBSMT system as the baseline for comparison. Ours performed better than Baseline by using only a small size of translation rules that is about one fourth of that of Baseline.

<table>
<thead>
<tr>
<th>System</th>
<th>Rule #</th>
<th>BLEU</th>
<th>RIBES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>35M</td>
<td>34.25</td>
<td>78.94</td>
</tr>
<tr>
<td>Ours</td>
<td>8.8M</td>
<td>34.87</td>
<td>79.25</td>
</tr>
</tbody>
</table>