Improved Statistical Machine Translation with Source Language Paraphrase

Chen Su, Yujie Zhang, Jin’an Xu, Zhen Guo
Beijing Jiaotong University
Outline

- Background
- Investigation & Research
- Decoding
- Experiment & Result
SMT system needs training data

- Parallel corpus
- Large-scale, Wide-cover

Existing problems

- It is difficult to obtain large scale parallel corpus
Out-Of-Vocabulary (OOV)

- SMT system didn’t do anything for OOVs, so they are retained in translation, which affect translation quality.

Now let me talk about Article II on Labor Law.

现在，让我们讨论一下关于劳动法的article II.
Lacking abundant candidate translations

- Generally speaking, phrase translation table can't cover all translation knowledge for every phrase, which causes the test sentences cannot be translated properly.

Now let me talk about Article II on Labor Law.

现在，让我们讨论一下关于劳动法的文章 II
Now let me talk about Article II on Labor Law.
Now let me talk about Article II on Labor Law.

Paraphrase
Now let me talk about Article II on Labor Law.

Now let me talk about Clause II on Labor Law.

现在，让我们讨论一下关于劳动法的条款 II
Framework: SMT with paraphrase
Related work

- Integrating paraphrase knowledge for SMT (Du Jinhua 2010)
  - Use lattice graph to denote input sentence's different paraphrases.
  - Du's work adopt heuristic method for estimate the weight of paraphrase. (Fixed weights)
Obtain paraphrases by pivoting with additional bilingual corpora

\[ \text{para}(e_2 | e_1) \approx \sum_{jp} p(e_2 | jp) \cdot p(jp | e_1) \]

\[ p(e_2 | jp) \approx \frac{\text{count}(e_2, jp)}{\sum_{e_2} \text{count}(e_2, jp)} \]

\[ p(jp | e_1) \approx \frac{\text{count}(jp, e_1)}{\sum_{jp} \text{count}(jp, e_1)} \]
Investigation (1/5)

Purpose

• How the coverage of OOV is improved with paraphrase
• How the translation is improved with paraphrase

Experimental data

• NTCIR-10 English-Chinese corpus
  • Used for acquiring translation knowledge
• NTCIR-10 English-Japanese corpus
  • Used for acquiring paraphrase knowledge
Investigation (2/5)

Experiment & Evaluation for coverage of OOV

- Divide the training data into 3 different parts with different scale, and then acquire translation knowledge.
- Compare coverage of N-gram before and after adding paraphrase.

<table>
<thead>
<tr>
<th>Number of N-gram on Testing data</th>
<th>Coverage (Phrase translation → Phrase translation with paraphrase)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10K Training Data</td>
</tr>
<tr>
<td>1元(6,274)</td>
<td>77.19% → 89.97%</td>
</tr>
<tr>
<td>2元(28,993)</td>
<td>35.57% → 67.52%</td>
</tr>
<tr>
<td>3元(42,937)</td>
<td>12.97% → 37.11%</td>
</tr>
<tr>
<td>4元(46,974)</td>
<td>4.20% → 14.94%</td>
</tr>
<tr>
<td>5元(47,316)</td>
<td>1.50% → 5.55%</td>
</tr>
<tr>
<td>6元(46,389)</td>
<td>0.55% → 2.12%</td>
</tr>
<tr>
<td>7元(44,918)</td>
<td>0.24% → 0.81%</td>
</tr>
</tbody>
</table>
Experiment: Coverage of correct translation

- **Definition:**
  - *Similarity*: longest common subsequence between translation and reference, which is normalized by length of reference translation. (the unit is character)
  - *Ideal translation*: the candidate translation with the highest similarity score.

- **Objects:** evaluate changes of the similarity between ideal translation and reference without/with paraphrase.

- **Methods:** search the ideal translation with CKY algorithm.
Longest common subsequence between English’s phrase $e_i^j$’s ideal translation and reference $c_i^m$

$$f(e_i^j, c_i^m) = \max \begin{cases} 
\text{Length}(c_i^m) \\
 f(e_i^k, c_i^n) + f(e_i^j, c_i^m) \\
 f(e_i^k, c_i^m) + f(e_i^j, c_i^n) \\
 f(e_{i+1}^j, c_i^m) \\
 f(e_{i-1}^j, c_i^m) \\
 f(e_i^j, c_{i+1}^m) \\
 f(e_i^j, c_{m-1}^m) \\
\end{cases}$$

if $e_i^j \rightarrow c_i^m$ is existing in phrase table (PT)

(i $< k < j$, $l < n < m$) monotone order

(i $< k < j$, $l < n < m$) swap order

if $e_{i+1}^j \rightarrow \text{null}$ is existing in PT

if $e_{j-1}^j \rightarrow \text{null}$ is existing in PT

if $c_i^{l+1} \rightarrow \text{null}$ is existing in PT

if $c_i^m \rightarrow \text{null}$ is existing in PT

$\text{Similarity} (c, c_{\text{ref}}) = \sum_{s=1}^{S} f(e_s^c, c_s^c) / \sum_{s=1}^{S} \text{Length}(c_s^c)$

$S$ is the number of sentences of the testing data.
Changes of the similarity without/with paraphrase

<table>
<thead>
<tr>
<th>Translation resource</th>
<th>Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10K Training data</td>
</tr>
<tr>
<td>Phrase translation table</td>
<td>82.82%</td>
</tr>
<tr>
<td>Phrase translation table + paraphrase knowledge</td>
<td>92.46%</td>
</tr>
</tbody>
</table>
SMT decoding algorithm employing paraphrase (1/2)

- Decoding for maximum-entropy SMT model
  \[ \hat{c} = \arg \max_c \{ \Pr(c|e_1) \} = \arg \max_c \{ \sum_{m=1}^{M} \lambda_m h_m(c, e_1) \} \]

- Features about phrase via paraphrase
  
  - (In order/Reverse) phrase translation features
    \[ \hat{h}_{Tran}(c, e_1) = \log \hat{p}(c|e_1) = \log \left[ \text{para}(e_2|e_1)^{\alpha_1} \cdot p(c|e_2) \right] \]
    \[ \hat{h}_{VerTran}(c, e_1) = \log \hat{p}(e_1|c) = \log \left[ \text{para}(e_1|e_2)^{\alpha_2} \cdot p(e_2|c) \right] \]

  - (In order/Reverse) lexical translation features
    \[ \hat{h}_{Lex}(c, e_1) = \log \hat{Lex}(c|e_1) = \log \left[ \text{para}(e_2|e_1)^{\alpha_3} \cdot Lex(c|e_2) \right] \]
    \[ \hat{h}_{VerLex}(c, e_1) = \log \hat{Lex}(e_1|c) = \log \left[ \text{para}(e_1|e_2)^{\alpha_4} \cdot Lex(e_2|c) \right] \]
SMT decoding algorithm employing paraphrase (2/2)

Formula transformation:

- two new paraphrase features are added.

\[
\lambda_{Tran} \cdot \hat{h}_{Tran}(c, e_1) + \lambda_{Lex} \cdot \hat{h}_{Lex}(c, e_1) \\
= \lambda_{Tran} \log p(c \mid e_2) + \lambda_{Lex} \log \text{Lex}(c \mid e_2) + (\lambda_{Tran} \cdot \alpha_1 + \lambda_{Lex} \cdot \alpha_3) \log \text{para}(e_2 \mid e_1) \\
= \lambda_{Tran} \cdot h_{Tran}(c, e_2) + \lambda_{Lex} \cdot h_{Lex}(c, e_2) + \lambda_{Para} \cdot h_{Para}(e_1, e_2)
\]

\[
\lambda_{VerTran} \cdot \hat{h}_{VerTran}(c, e_1) + \lambda_{VerLex} \cdot \hat{h}_{VerLex}(c, e_1) \\
= \lambda_{VerTran} \cdot h_{VerTran}(c, e_2) + \lambda_{VerLex} \cdot h_{VerLex}(c, e_2) + \lambda_{VerPara} \cdot h_{VerPara}(e_1, e_2)
\]

Use MERT for optimizing model weight.
Data

- NTCIR-10 English-Chinese corpus used for training SMT system
- NTCIR-10 English-Japanese corpus used for acquiring paraphrase

SMT system for comparison

- Baseline: traditional phrase-based SMT
- Du System: traditional phrase-based SMT with fixed weight of paraphrase lattice
- Our System: traditional phrase-based SMT with paraphrase features
<table>
<thead>
<tr>
<th>SMT system</th>
<th>10K training data</th>
<th>100K training data</th>
<th>1M training data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>35.69</td>
<td>40.39</td>
<td>44.16</td>
</tr>
<tr>
<td>Du System</td>
<td>36.72(+1.03)</td>
<td>40.68(+0.29)</td>
<td>43.43(-0.73)</td>
</tr>
<tr>
<td>Our System</td>
<td>37.09(+1.40)</td>
<td>40.42(+0.30)</td>
<td>44.48(+0.32)</td>
</tr>
</tbody>
</table>
Conclusion

- Integrated paraphrase into SMT
  - We redesigned the paraphrase as features in SMT decoding algorithm.
- Investigated the effect of paraphrase in
  - Coverage of OOV
  - Coverage of correct translation
- Evaluated the SMT system performance
  - The improvement was proved in BLEU.
Thank you!