Improving SMT by Paraphrasing the Training Data

Francis Bond (NICT)
Eric Nichols (NAIST), Darren Scott Appling (Georgia Tech),
Michael Paul (NICT)
<bond@ieee.org>

October 20, 2008
Motivation

➢ SMT quality improves with more (in-domain) data
   ➢ +2.5% BLEU if you double the bitext
   ➢ +0.5% BLEU for target)

➢ But bitext is rare (and expensive to make) which makes SMT/EBMT hard to deploy

➢ How can we cheaply make more bitext?
   ➢ Translate more — expensive
   ➢ Find more — great if there is some
   ➢ Extend existing data — our approach
(1)  この こと から 、 会社 に は 事故 の 責任 が 無い こと に なる。

It follows from this that the company is not responsible for the accident.

It follows that the company isn’t responsible for the accident from this.

It follows that the company is not responsible for the accident from this.

That the company isn’t responsible for the accident follows from this.
Paraphrasing

➢ There are often multiple ways of saying the same thing
  ➢ Almost always they differ in some nuance
  ➢ But sometimes these differences are negligible

➢ Two main kinds of paraphrase
  ➢ Structural — different word order, different function words
    We can do this by parsing and generating (using HPSG)
  ➢ Lexical — different (open class) word choice
    We can do this with WSD and then sense substitution

➢ Need to do together for full generality
  
  *I like pears. ↔ Pears please me.*
Previous Work

- Paraphrase by finding translation equivalents (Callison-Burch et al., 2006)
- SMT with language A and C
- find places where multiple $A_1$, $A_2$ links to one C
- SMT between A and B
  * If $A_1$-$B$ is not in phrase table, try to replace by $A_2$-$B$

- Example: *used, use, spent, utilize. to use (Es: usado)*

- Raises BLEU 1% for small training data, saturates at 160K; good for unknown words/phrases
Previous Work (2)

- Noun phrase rewriting (Nakov, 2008)
  
  2) of members of the Irish parliament
  of irish parliament members
  of irish parliament’s members

- Explicit Paraphrasing to make translation easier
  
  - Source rewriting in RBMT (Shirai et al., 1993)
    * Dependency tree reordering using rules
  
  - Re-ordering for SMT (Komachi et al., 2006)
    * Chunk reordering using rules
Our Approach

➢ Attempt to build new bitext pairs

➢ replace one side with grammatical, semantically equivalent variants

➢ Parse to a structural meaning representation:
  MRS (Minimal Recursion Semantics)
  rank with stochastic model

➢ Generate from 1-best MRS
  rank with stochastic model
  select top $n$ (up to 10)
Using the English Resource Grammar (Flickinger, 2000)

Parse with PET; generate with LKB

Various kinds of variation:

- Phrase order
- Closed class words: everyone, everybody
- Contractions: going to vs gonna
- Numbers: three vs 3
- Correction: I read the the book vs I read the book
- Punctuation
Why is this useful?

➢ Consider: *going to* (main verb) vs *gonna* (auxiliary)

➢ *I am going to the store* (200)

➢ *I am going to cry* (600) vs *I am gonna cry*

➢ *I am gonna cry* (9) vs *I am going to cry*

➢ Paraphrasing disambiguates

➢ Paraphrasing helps with scarcity

➢ More constrained than phrase table paraphrasing
Experiment

➢ Paraphrasing (English)
   ERG grammar, pet parser, lkb generator
   DELPH-in: Deep Linguistic Processing with HPSG Initiative

➢ SMT system: Moses (Koehn et al., 2007)
   replacing giza++ with mgiza

➢ Corpora
  ➢ Tanaka Corpus (EJ) (2005 version)
    147,190 training, 4,500 dev, 4,500 test
  ➢ IWSLT corpus (EJ) (2005 version)
    42,699 training, 2,108 dev, 500 test (different set)
Everybody often goes to the the movies.

\[
\begin{align*}
\langle h_1, & \quad h_3: \text{person} (\text{ARG0 } x_4 \{\text{PERS 3, NUM sg}\}), \\
& \quad h_5: \text{every}_\text{q} (\text{ARG0 } x_4, \text{RSTR } h_6, \text{BODY } h_7), \\
& \quad h_8: \text{often}_\text{a}_\text{1} (\text{ARG0 } e_9 \{\text{TENSE untensed}\}, \text{ARG1 } e_2 \{\text{TENSE pres}\}), \\
& \quad h_8: \text{go}_\text{v}_\text{1} (\text{ARG0 } e_2, \text{ARG1 } x_4), \\
& \quad h_8: \text{to}_\text{p} (\text{ARG0 } e_{10} \{\text{TENSE untensed}\}, \text{ARG1 } e_2, \text{ARG2 } x_{11}) \\
& \quad h_{12}: \text{the}_\text{q} (\text{ARG0 } x_{11}, \text{RSTR } h_{14}, \text{BODY } h_{13}), \\
& \quad h_{15}: \text{movie}_\text{n}_\text{of} (\text{ARG0 } x_{11}, \text{ARG1 } i_{16} \{\text{SF prop}\}) \\
\{ h_6 =_q h_3, h_{14} =_q h_{15} \} \rangle
\end{align*}
\]
**Paraphrases**

*Everybody often goes to the the movies.*

- *Everyone often goes to the movies.* 7.7
- *Everybody often goes to the movies.* 7.7
- *Everyone goes often to the movies.* 0.5
- *Everybody goes often to the movies.* 0.5
- *Everyone goes to the movies often.* -0.3
- *Everybody goes to the movies often.* -0.3

For the Tanaka Corpus, 83.4% could be paraphrased.

<table>
<thead>
<tr>
<th># distinct paraphrases</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of sentences</td>
<td>53.4</td>
<td>31.2</td>
<td>21.2</td>
<td></td>
<td>1.1</td>
</tr>
</tbody>
</table>
(d)istributed: rotate between the original sentence and each paraphrase until the data has been padded out

(f)irst: after all paraphrases have been used, the first (original) sentence is repeated to pad out the data

(v)arying: add just the paraphrases (always does worse)

$$
\begin{array}{c|cccccc}
\text{d} & e_0 & e_1 & e_2 & e_0 & e_1 \\
\text{f} & e_0 & e_1 & e_2 & e_0 & e_0 \\
\text{v} & e_0 & e_1 & e_2 \\
\end{array}
$$

Table 1: Paraphrase distributions ($n = 4, m = 2$)
## Results (TC-EJ)

<table>
<thead>
<tr>
<th>Lang Pair</th>
<th>Corpus</th>
<th>Paraphrases Added</th>
<th>Bleu</th>
<th>Variance</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>EJ</td>
<td>Tanaka Corpus</td>
<td>0</td>
<td>25.96</td>
<td>±0.71</td>
<td>-</td>
</tr>
<tr>
<td>EJ</td>
<td>Tanaka Corpus</td>
<td>d.2</td>
<td>26.10</td>
<td>±0.74</td>
<td>+0.14</td>
</tr>
<tr>
<td>EJ</td>
<td>Tanaka Corpus</td>
<td>d.4</td>
<td>26.25</td>
<td>±0.71</td>
<td>+0.29</td>
</tr>
<tr>
<td>EJ</td>
<td>Tanaka Corpus</td>
<td>d.6</td>
<td>26.63</td>
<td>±0.72</td>
<td>+0.67</td>
</tr>
<tr>
<td>EJ</td>
<td>Tanaka Corpus</td>
<td>d.8</td>
<td>26.16</td>
<td>±0.71</td>
<td>+0.20</td>
</tr>
<tr>
<td>EJ</td>
<td>Tanaka Corpus</td>
<td>f.4</td>
<td>26.28</td>
<td>±0.73</td>
<td>+0.32</td>
</tr>
<tr>
<td>EJ</td>
<td>Tanaka Corpus</td>
<td>f.6</td>
<td>26.13</td>
<td>±0.68</td>
<td>+0.17</td>
</tr>
<tr>
<td>EJ</td>
<td>Tanaka Corpus</td>
<td>f.8</td>
<td>25.83</td>
<td>±0.65</td>
<td>-0.13</td>
</tr>
<tr>
<td>Lang Pair</td>
<td>Corpus</td>
<td>Paraphrases Added</td>
<td>Bleu</td>
<td>Variance</td>
<td>Delta</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>--------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>JE</td>
<td>Tanaka Corpus</td>
<td>0</td>
<td>18.75</td>
<td>±0.82</td>
<td>-</td>
</tr>
<tr>
<td>JE</td>
<td>Tanaka Corpus</td>
<td>d.2</td>
<td>19.09</td>
<td>±0.74</td>
<td>+0.34</td>
</tr>
<tr>
<td>JE</td>
<td>Tanaka Corpus</td>
<td>d.4</td>
<td>18.42</td>
<td>±0.79</td>
<td>-0.33</td>
</tr>
<tr>
<td>JE</td>
<td>Tanaka Corpus</td>
<td>d.6</td>
<td>18.71</td>
<td>±0.83</td>
<td>-0.04</td>
</tr>
<tr>
<td>JE</td>
<td>Tanaka Corpus</td>
<td>d.8</td>
<td>18.90</td>
<td>±0.77</td>
<td>+0.15</td>
</tr>
<tr>
<td>JE</td>
<td>Tanaka Corpus</td>
<td>f.4</td>
<td>18.92</td>
<td>±0.81</td>
<td>+0.17</td>
</tr>
<tr>
<td>JE</td>
<td>Tanaka Corpus</td>
<td>f.6</td>
<td>19.02</td>
<td>±0.80</td>
<td>+0.27</td>
</tr>
<tr>
<td>JE</td>
<td>Tanaka Corpus</td>
<td>f.8</td>
<td>19.19</td>
<td>±0.82</td>
<td>+0.44</td>
</tr>
</tbody>
</table>
EJ: Significant improvements overall (up to +.67)

JE: Some improvement, some degradation

Hard to decide how many paraphrases to use

Should probably try to select good paraphrases

More consistent, but worse absolute, results with older Moses
EJ: less improvement over all (up to +.61)

JE: non-significant improvement

Probably because of more paraphrases in the original corpus

More consistent, but worse absolute, results with older Moses
Why we think it works

➢ We can improve precision for most tasks by adding another knowledge source

➢ Paraphrasing with the HPSG grammar allows the system to generalize over variants

➢ May also make alignment easier for some word orders

➢ But we do not generate all variants

➢ And we do not rank so well (Im from tourism corpus)
Future Work (1)

➤ Explicitly weight the equivalents

➤ $n$ paraphrases with weight $1/n$ or $p(s)$

➤ Also do Japanese (using Jacy)

➤ Retrain parse/generation ranking models (need treebanks)

➤ Only take paraphrases with score above a threshold
Future Work (2)

➢ Add an En-EN transfer step
  ➢ NP rewriting
  ➢ Idiom ↔ literal
  ➢ Active ↔ passive
  ➢ Lexical paraphrases

➢ Combine with a Ja-Ja transfer step
  ➢ Statement ↔ question
  ➢ Positive ↔ negative

➢ Use to normalize (All going to to gonna?)
Resource Availability

We have release the paraphrased corpus. It should be possible to reproduce the results.

➢ Paraphrased Tanaka Corpus

➢ Tanaka Corpus (2005 version)
   (2008 version ≈ 12% corrected) — coming soon

➢ English grammar, parser generator (ERG, pet, lkb)
   DELPH-IN: www.delph-in.net

➢ SMT system (Moses, mgiza++ and dependencies)

➢ Ubuntu NLP repositories (packaged by Eric Nichols)
We can improve the quality of SMT-based translations by automatically creating more training data.

We create more training data by paraphrasing one side. Parse to semantic representation (MRS) select the most plausible interpretation. Generate all sentences with the same meaning select the ($n$-most) fluent sentences.

The resources needed to do this are publicly available.
Comparison

➢ Resources needed

➢ A different bitext (Callison-Burch et al., 2006)
➢ An HPSG grammar (En, Ja, De, No, Es, Pt, Gr)

➢ Effectiveness

➢ Good for small corpus (Callison-Burch et al., 2006)
➢ Small constant improvement (source language paraphrase)

➢ No reason not to combine the two methods
References


