MT System Combination by Confusion Forest

Taro Watanabe and Eiichiro Sumita
NICT
MT System Combination
MT System Combination

- Better translation by combining multiple system outputs:
  - Sentence selection (Nomoto, 2004; etc.)
  - Phrasal combination (Frederking and Nirenburg, 1994; etc.)
  - Word level combination (Bangalore et al., 2001; Matusov et al., 2006; etc.)
MT System Combination

- Better translation by combining multiple system outputs:
  - Sentence selection (Nomoto, 2004; etc.)
  - Phrasal combination (Frederking and Nirenburg, 1994; etc.)
  - Word level combination (Bangalore et al., 2001; Matusov et al., 2006; etc.)
- This Work: Syntactic combination, not word-wise combination
Confusion Network
Confusion Network

I saw the forest
I walked the blue forest
I saw the green trees
the forest was found
I saw the forest
I walked the blue forest
I saw the green trees
the forest was found

• State-of-the-art: Confusion Network
• Choose a skeleton, compute word alignment against the skeleton
  • Edit-distance-based alignment (TER etc.) (Sim et al., 2007)
  • Model-based alignment (GIZA++ etc.) (Matsov et al., 2006)
Confusion Network

★ I saw the forest
I walked the blue forest
I saw the green trees
the forest was found

• State-of-the-art: Confusion Network

• Choose a skeleton, compute word alignment against the skeleton

  • Edit-distance-based alignment (TER etc.) (Sim et al., 2007)

  • Model-based alignment(GIZA++ etc.) (Matsov et al., 2006)
Confusion Network

★
I saw the forest
I walked the blue forest
I saw the green trees
the forest was found

• State-of-the-art: Confusion Network

• Choose a skeleton, compute word alignment against the skeleton
  • Edit-distance-based alignment (TER etc.) (Sim et al., 2007)
  • Model-based alignment (GIZA++ etc.) (Matsov et al., 2006)
Confusion Network

• Construct a network with each arc representing alternative translation
• Best path = Best translation
• Syntactically different language pairs: i.e. active/passive voices
• Spurious insertion/repetition due to alignment error
• Incremental alignment/construction + merge multiple networks into one (Rosti et al., 2008)
Confusion Network

- Construct a network with each arc representing alternative translation
- Best path = Best translation
- Syntactically different language pairs: i.e. active/passive voices
- Spurious insertion/repetition due to alignment error
- Incremental alignment/construction + merge multiple networks into one (Rosti et al., 2008)
Confusion Network

- Construct a network with each arc representing alternative translation
- Best path = Best translation
- Syntactically different language pairs: i.e. active/passive voices
- Spurious insertion/repetition due to alignment error
- Incremental alignment/construction + merge multiple networks into one (Rosti et al., 2008)
Confusion Forest

- Compactly represent multiple parses by sharing nodes
- Represented by “hypergraph”
Confusion Forest

\[ e_1 = \langle VP^{@2}, \{ VBD^{@3}, VP^{@4} \} \rangle \]

- Compactly represent multiple parses by sharing nodes
- Represented by “hypergraph”
Confusion Forest

- Compactly represent multiple parses by sharing nodes
- Represented by “hypergraph”
Confusion Forest

$e_1 = \langle \text{VP}^@2, \{ \text{VBD}^@3, \text{VP}^@4 \} \rangle$

$e_2 = \langle \text{VP}^@2, \{ \text{VBD}^@2.1, \text{NP}^@2.2 \} \rangle$

- Compactly represent multiple parses by sharing nodes
- Represented by “hypergraph”
Confusion Forest

- Compactly represent multiple parses by sharing nodes
- Represented by “hypergraph”
Rule Extraction

- Parse each system output by a parser
- Extract rules from parsed trees: local grammar
• Parse each system output by a parser
• Extract rules from parsed trees: local grammar
• Parse each system output by a parser

• Extract rules from parsed trees: local grammar
Rule Extraction

- Parse each system output by a parser
- Extract rules from parsed trees: local grammar
Generation by Earley

Scan:

\[
\begin{align*}
[X \rightarrow \alpha \cdot x\beta, h] : u \\
[X \rightarrow \alpha x \cdot \beta, h] : u
\end{align*}
\]

Predict:

\[
\begin{align*}
[X \rightarrow \alpha \cdot Y\beta, h] \\
[Y \rightarrow \bullet \gamma, h + 1] : u
\end{align*}
\]

\[
Y \xrightarrow{u} \gamma \in G, h < H
\]

Complete:

\[
\begin{align*}
[X \rightarrow \alpha \cdot Y\beta, h] : u \\
[Y \rightarrow \gamma \bullet, h + 1] : v
\end{align*}
\]

\[
[X \rightarrow \alpha Y \cdot \beta, h] : u \otimes v
\]

- Generation from the extracted grammar
- Scanning always succeed: constraint by height
Generation by Earley
Generation by Earley

$S^@\varepsilon$
Generation by Earley

I walked

I the forest

NP

S

VP

PRP DT NN

NP@1

NP@2

VP@2
Generation by Earley

S@ε

NP@1  VBD@3  VP@4

PRP  DT  NN  was  VBN

I  the  forest  found

VP@2

VBD@2.1  NP@2.2
Generation by Earley

S@ε

NP@1  VBD@3  VP@4
PRP  DT  NN  was  VBN
I  the  forest  found

VP@2

VBD@2.1
walked  saw  NP@2.2

NP@2.2

DT@2.2.1  NN@2.2.2
the  blue  green  the  forest

DT  JJ

the  blue  green

forest  trees
Spurious Ambiguity

- Memorize the (partial) tree structures in each node
- Employ the sequence of Earleye state as a node
- Horizontal/Vertical Markovization (Klein and Manning, 2003)
Spurious Ambiguity

- Memorize the (partial) tree structures in each node
- Employ the sequence of Earleye state as a node
- Horizontal/Vertical Markovization (Klein and Manning, 2003)
Spurious Ambiguity

- Memorize the (partial) tree structures in each node
- Employ the sequence of Earleye state as a node
- Horizontal/Vertical Markovization (Klein and Manning, 2003)
Spurious Ambiguity

- Memorize the (partial) tree structures in each node
- Employ the sequence of Earlye state as a node
- Horizontal/Vertical Markovization (Klein and Manning, 2003)
Spurious Ambiguity

- Memorize the (partial) tree structures in each node
- Employ the sequence of Earlye state as a node
- Horizontal/Vertical Markovization (Klein and Manning, 2003)
Spurious Ambiguity

- Memorize the (partial) tree structures in each node
- Employ the sequence of Earleye state as a node
- Horizontal/Vertical Markovization (Klein and Manning, 2003)
Forest Reranking

\[ \hat{d} = \arg \max_{d \in D} \mathbf{w}^\top \cdot \mathbf{h}(d, F) \]

- Choose the best derivation \( d \) among all possible derivations \( D \) in a forest \( F \)
- Terminal yield of the best derivation = the best translation
- Approximately apply non-local features (ngram language models) by Cube Pruning (Huang and Chiang, 2007)
- Efficient \( k \)-best by Algorithm 3 (Huang and Chiang, 2005)
Experiments

- WMT10 System Combination Task
- Czech, German, Spanish, French → English
- tune/test: 455/2,034 sentences

<table>
<thead>
<tr>
<th></th>
<th>cz-en</th>
<th>de-en</th>
<th>es-en</th>
<th>fr-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>systems</td>
<td>6</td>
<td>16</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>tune</td>
<td>10.6K</td>
<td>10.9K</td>
<td>10.9K</td>
<td>11.0K</td>
</tr>
<tr>
<td>test</td>
<td>50.5K</td>
<td>52.1K</td>
<td>52.1K</td>
<td>52.4K</td>
</tr>
</tbody>
</table>
Systems
Systems

- CF: Stanford parser + “cicada” (a hypergraph-based toolkit based on SEMIring parsing framework)
- CN: Single network by merging multiple networks + conversion into hypergraph by lattice parsing
Systems

• CF: Stanford parser + “cicada” (a hypergraph-based toolkit based on SEMIrng parsing framework)

• CN: Single network by merging multiple networks + conversion into hypergraph by lattice parsing

• features: tuned by hypergraph-MERT(Kumar et al. 2009)
  • Language Models, # of terminals, # of hyperedges
  • # of rules in a derivation originally in $n_{th}$ system output
  • BLEUs by treating each system output as a reference translation
  • Network distance (only used for CN)
## BLEU

<table>
<thead>
<tr>
<th></th>
<th>cz-en</th>
<th>de-en</th>
<th>es-en</th>
<th>fr-en</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>system min</strong></td>
<td>14.09</td>
<td>15.62</td>
<td>21.79</td>
<td>16.79</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>23.44</td>
<td>24.10</td>
<td>29.97</td>
<td>29.17</td>
</tr>
<tr>
<td><strong>CN</strong></td>
<td>23.70</td>
<td>24.09</td>
<td>30.45</td>
<td>29.15</td>
</tr>
<tr>
<td>CF,v=∞,h=∞</td>
<td>24.13</td>
<td>24.18</td>
<td>30.41</td>
<td>29.57</td>
</tr>
<tr>
<td>CF,v=∞,h=2</td>
<td>24.14</td>
<td>24.58</td>
<td>30.52</td>
<td>28.84</td>
</tr>
<tr>
<td>CF,v=∞,h=1</td>
<td>24.01</td>
<td>23.91</td>
<td>30.46</td>
<td>29.32</td>
</tr>
</tbody>
</table>
## Oracle BLEU

<table>
<thead>
<tr>
<th></th>
<th>cz-en</th>
<th>de-en</th>
<th>es-en</th>
<th>fr-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>rerank</td>
<td>29.40</td>
<td>32.32</td>
<td>36.83</td>
<td>36.59</td>
</tr>
<tr>
<td>CN</td>
<td>38.52</td>
<td>34.97</td>
<td>47.65</td>
<td>46.37</td>
</tr>
<tr>
<td>CF,v=∞,h=∞</td>
<td>30.51</td>
<td>34.07</td>
<td>38.69</td>
<td>38.94</td>
</tr>
<tr>
<td>CF,v=∞,h=2</td>
<td>30.61</td>
<td>34.25</td>
<td>38.87</td>
<td>39.10</td>
</tr>
<tr>
<td>CF,v=∞,h=1</td>
<td>31.09</td>
<td>34.65</td>
<td>39.27</td>
<td>39.51</td>
</tr>
</tbody>
</table>
## Hypegraph size

<table>
<thead>
<tr>
<th></th>
<th>cz-en</th>
<th>de-en</th>
<th>es-en</th>
<th>fr-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>2,222.68</td>
<td>47,231.20</td>
<td>2,932.24</td>
<td>11,969.40</td>
</tr>
<tr>
<td>CF, v=∞, h=1</td>
<td>230.08</td>
<td>540.03</td>
<td>262.30</td>
<td>386.79</td>
</tr>
<tr>
<td>CF, v=5, h=1</td>
<td>254.45</td>
<td>651.10</td>
<td>302.01</td>
<td>477.51</td>
</tr>
<tr>
<td>CF, v=4, h=1</td>
<td>286.01</td>
<td>802.79</td>
<td>349.21</td>
<td>575.17</td>
</tr>
</tbody>
</table>

- Average # of hyperedges
- (rough) estimates for speed
Conclusion

• System combination by Confusion Forest which employs syntactic distance, not word-level distance

• Forest construction by the grammar extracted from system outputs

• Parser: assign tree structure to the similar expressions

• Compact date structure + comparable performance against Confusion Network

• Future work

• Syntactic features