Integration of a Multilingual Preordering Component into a Commercial SMT Platform

Anita Ramm, Riccardo Superbo, Dimitar Shterionov, Tony O’Dowd, Alexander Fraser

IMS, University of Stuttgart
KantanMT.com
CIS, University of Munich

May 29th, 2017
Outline

1. Word Order in SMT
2. Our Approach
3. Experiments and Evaluation
   - SMT
   - NMT
4. Summary and Future Work
Outline

1. Word Order in SMT
2. Our Approach
3. Experiments and Evaluation
   - SMT
   - NMT
4. Summary and Future Work
Word order in Statistical Machine Translation

- Translation of a SL word needs to be placed in a TL-specific position
  - Fluency of the translations
- Problematic for languages with considerably different syntactic structure, e.g.:
  - EN: ...that I read a book yesterday.
  - DE: ...dass ich gestern ein Buch gelesen_{part} habe_{finV}.

- Particularly, the long-range reorderings lead to false placement or omission of TL words
  - Negative impact on the adequacy and fluency of the generated translations!
Reordering in SMT

- Many different approaches to handle reordering problems within SMT (cf. [Bisazza & Federico, 2016])
- One of the simplest, yet most effective approaches: preordering
  
  EN: ...that I a book yesterday read.
  
  DE: ...dass ich gestern ein Buch gelesen habe.

- Reordering may be based on either automatically derived rules or hand-crafted rules
- Reordering is carried out on parsed SL sentences, on the POS-tagged or non-processed SL corpus
1. Word Order in SMT

2. Our Approach

3. Experiments and Evaluation
   - SMT
   - NMT

4. Summary and Future Work
Reordering for a commercial SMT platform

- Multilingual preordering component for English→Japanese/German/Chinese SMT
- Rule-based reordering based on hand-crafted rules
  ⇒ improvement already shown by previous research
  ⇒ easily adapt to the client’s needs
- Stand-alone component used as a part of corpus preprocessing
  ⇒ ensures backward compatibility
- Generic implementation independent of the used parsing software
  ⇒ allows for usage of different parsers
Our Approach

Languages under consideration

1 English→German:
   - Rule set based on [Gojun & Fraser, 2012]
   - Movements of the verbs and negation to capture different positions of the verbs between EN and DE
   - Total of 9 reordering rules

2 English→Japanese:
   - Rule set based on [Lee et al., 2012]
   - Movements of various sentence constituents
   - Total of 7 reordering rules
   - and 1 insertion rule (to cope with null subjects in JA)

3 English→Chinese:
   - Rule set developed based on [Wang et al., 2007] and [Wu, 2016]
   - Movements of various sentence constituents
     ⇒ doesn’t work!
   - Exclusive reordering of NP-PPs and ofPPs
     (2 rules apply independently)
Our Approach
Architecture overview

English input sentence

The letter that I am writing is almost finished

PARSING
- tokenize
- parse

Tsurgeon

REORDERING

rules-en-ja.xml
rules-en-de.xml
rules-en-zh.xml

NP
SBAR NP
IN S
NP VP
VBZ VP
VP

S
NP
SBAR
IN
S
NP
VP
VBZ
VP

VP
NP=vp > VP $+ SBAR=sbar
move np $− sbar

REORDERING RULE

that I am writing The letter is almost finished

READ OUT

that I am writing The letter is almost finished
Our Approach

Architecture overview

Parsing incl. tokenisation

The letter that I am writing is almost finished

PARSING

- tokenise
- parse

Tsurgeon

- rules-en-ja.xml
- rules-en-de.xml
- rules-en-zh.xml

REORDERING

NP
SBAR NP
IN S
NP VP
VBZ VP
VP

The letter

that I am writing

is almost finished

rules−en−ja.xml

rules−en−de.xml

rules−en−zh.xml

REORDERING RULE

NP= np > NP $ + SBAR=sbar
move np $ - sbar

READ OUT

that I am writing The letter is almost finished
Our Approach
Architecture overview

Reordering based on Tsurgeon

The letter that I am writing is almost finished

PARSING
- tokenize
- parse

REORDERING
- REORDERING RULE
  - move np $− sbar
    - NP=np > NP $+ SBAR=sbar

rules−en−ja.xml
rules−en−de.xml
rules−en−zh.xml

Tsurgeon

The letter that I am writing is almost finished

that I am writing The letter is almost finished

and hand-crafted tree modification rules
Our Approach
Architecture overview

The letter that I am writing is almost finished

PARSING

parse
tokenise

Tsurgeon

REORDERING

rules-en-ja.xml
rules-en-de.xml
rules-en-zh.xml

NP
SBAR NP
IN S
NP VP
VBZ VP
VP

The
letter
that I am
writing
is
almost
finished

S
NP
SBAR
IN
S
NP
VBZ
VP

The letter
that
am
writing

rules-en-ja.xml
rules-en-de.xml
rules-en-zh.xml

READ OUT

NP=np > NP $+ SBAR=sbar
move np $− sbar

REORDERING RULE

that I am writing The letter is almost finished

Read out the reordered sentence
Processing steps

For each sentence in the input file:

1. Tokenise/parse the input sentence
   (constraints: 5 < len(sentence) < 60, no special characters)
2. Modify the parse tree (i.e., perform reordering)
3. Read out the sentence from the modified parse tree

After processing the entire input file, continue with tokenisation, lowercasing, etc.

Reordering

- **Tsurgeon-based modifications** of the parse trees
  [Levy & Andrew, 2006]
- Rules are defined following **standardised Tsurgeon syntax**
- Rule sets can easily be changed/adapted/extended
- Reordering is invoked only if the reordering rules are given
Our Approach

Implementation details

- Parser runs as a web server
  ⇒ makes the preordering process independent of the used parser

- Tsurgeon is modified to enable sentence-wise processing and to avoid infinite loops during tree modifications
  ⇒ ensures that no SL sentences are lost during preordering

- Processing parallelised with GNU parallel [Tange, 2011]
  ⇒ leads to a substantially lower reordering time
  (e.g., reordering of 5,000 segments with the Stanford Shift-Reduce (SR) parser run on 8 cores → the serial implementation took 263.24s, whereas the parallel implementation took 46.10s (5.7 times faster!))
Outline

1. Word Order in SMT
2. Our Approach
3. Experiments and Evaluation
   - SMT
   - NMT
4. Summary and Future Work
Experiments and Evaluation
Experimental setup for SMT

Training data (legal domain)

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>Train</th>
<th>Tune</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN→DE</td>
<td>1,018,738</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>EN→JA</td>
<td>213,592</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>EN→ZH</td>
<td>387,275</td>
<td>500</td>
<td>500</td>
</tr>
</tbody>
</table>

SMT models

- Trained with Moses
- Tuned on 500 in-domain sentences using MERT
- Distortion limit = 6
- 5-gram LMs trained with the TL of the parallel corpus
- Word alignment: fast_align
Experiments and Evaluation

Evaluation of SMT

Processing time vs. Translation improvement

- Preordering tested using output of different parsers with the same set of reordering rules

<table>
<thead>
<tr>
<th>Source-Target</th>
<th>Baseline BLEU</th>
<th>t&lt;sub&gt;t&lt;/sub&gt;</th>
<th>SR BLEU</th>
<th>t&lt;sub&gt;r&lt;/sub&gt;</th>
<th>t&lt;sub&gt;t&lt;/sub&gt;</th>
<th>PCFG BLEU</th>
<th>t&lt;sub&gt;r&lt;/sub&gt;</th>
<th>t&lt;sub&gt;t&lt;/sub&gt;</th>
<th>BLLIP BLEU</th>
<th>t&lt;sub&gt;r&lt;/sub&gt;</th>
<th>t&lt;sub&gt;t&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN→DE</td>
<td>40.10</td>
<td>187</td>
<td>40.74</td>
<td>97</td>
<td>254</td>
<td>41.17</td>
<td>372</td>
<td>579</td>
<td>41.49</td>
<td>1279</td>
<td>1468</td>
</tr>
<tr>
<td>EN→JA</td>
<td>49.44</td>
<td>135</td>
<td>51.33</td>
<td>25</td>
<td>155</td>
<td>50.29</td>
<td>413</td>
<td>544</td>
<td>51.33</td>
<td>372</td>
<td>492</td>
</tr>
<tr>
<td>EN→ZH (PP-NP)</td>
<td>24.99</td>
<td>197</td>
<td>24.40</td>
<td>50</td>
<td>245</td>
<td>24.47</td>
<td>252</td>
<td>460</td>
<td>24.66</td>
<td>627</td>
<td>819</td>
</tr>
<tr>
<td>EN→ZH (ofPP)</td>
<td>24.99</td>
<td>197</td>
<td>25.09</td>
<td>49</td>
<td>240</td>
<td>25.22</td>
<td>269</td>
<td>464</td>
<td>25.05</td>
<td>633</td>
<td>820</td>
</tr>
</tbody>
</table>

⇒ BLEU improves for all parsers!

- Additional processing time when the fastest parser (SR) is used: 36% for EN→DE, 15% for EN→JA, 22-24% for EN-ZH

- Positive user feedback justifies longer processing time!
Experiments and Evaluation
Experimental setup for NMT

Training data
(legal domain)

<table>
<thead>
<tr>
<th></th>
<th>train</th>
<th>tune</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN→DE</td>
<td>1,018,738</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>EN→JA</td>
<td>213,592</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>EN→ZH</td>
<td>387,275</td>
<td>500</td>
<td>500</td>
</tr>
</tbody>
</table>

NMT models
- Trained with OpenNMT
- BPE for EN→DE
- Character-based segmentation for EN→JA/ZH
- Training time: max of 15 epochs
Automatic (BLEU) & human evaluation (A/B testing) of the NMT outputs:

<table>
<thead>
<tr>
<th>Language</th>
<th>BLEU Human</th>
<th>BLEU Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN-DE</td>
<td>38.26</td>
<td>49.2</td>
</tr>
<tr>
<td>EN-JA</td>
<td>67.66</td>
<td>–</td>
</tr>
<tr>
<td>EN-ZH (PP-NP)</td>
<td>27.65</td>
<td>36.9</td>
</tr>
<tr>
<td>EN-ZH (ofPP)</td>
<td>28.75</td>
<td>32.4</td>
</tr>
</tbody>
</table>

⇒ Preordering hurts NMT!
⇒ Confirmed both by automatic, as well as human evaluation
Baseline generates the DE verbs in correct positions
(see, e.g., [Bentivogli et al., 2016])

Preordering seems to have impact both on word order, as well as word choice, e.g.:

<table>
<thead>
<tr>
<th>EN</th>
<th>The Commission may, in any case, withdraw such products or substances in accordance with Article37(2).</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENr</td>
<td>The Commission may, in any case, such products or substances in accordance with Article37(2) <strong>withdraw</strong>.</td>
</tr>
<tr>
<td>B</td>
<td>Die Kommission kann in jedem Fall <strong>diese Produkte oder Stoffe</strong> gemäß Artikel37 Absatz2 zurückziehen.</td>
</tr>
<tr>
<td>R</td>
<td>Die Kommission <strong>kann</strong> in jedem Fall <strong>solche Erzeugnisse oder Stoffe</strong> gemäß Artikel37 Absatz2 zurückziehen.</td>
</tr>
<tr>
<td>REF</td>
<td>Kommission <strong>kann</strong> in jedem Fall <strong>solche Erzeugnisse oder Stoffe</strong> gemäß Artikel37 Absatz2 zurückziehen.</td>
</tr>
</tbody>
</table>

⇒ Further investigation of preordering for NMT is needed
1. Word Order in SMT

2. Our Approach

3. Experiments and Evaluation
   - SMT
   - NMT

4. Summary and Future Work
Summary and Future Work

We presented...

- a fast and customisable preordering component\(^1\) for \(\text{EN} \rightarrow \text{DE/JA/ZH}\)
- **positive impact** of preordering on translation quality for SMT
- **negative impact** of preordering on translation quality for NMT

\(^1\)A simplified version of the preordering component is freely available for research purposes: https://github.com/KantanLabs/KantanPreorder.
We presented...

- a fast and customisable preordering component\(^1\) for EN→DE/JA/ZH
- positive impact of preordering on translation quality for SMT
- negative impact of preordering on translation quality for NMT

### Future work for SMT

- Parallelisation with BLLIP parser
- Use of domain-specific parsers to ensure sufficient parsing quality

\(^1\)A simplified version of the preordering component is freely available for research purposes: https://github.com/KantanLabs/KantanPreorder.
Summary and Future Work

We presented...

- a fast and customisable preordering component\(^1\) for EN→DE/JA/ZH
- positive impact of preordering on translation quality for SMT
- negative impact of preordering on translation quality for NMT

Future work for SMT

- Parallelisation with BLLIP parser
- Use of domain-specific parsers to ensure sufficient parsing quality

Future work for NMT

- Further investigation of the impact of the preordering on NMT
- Multi-source approach to combine reordered and non-reordered SL inputs

\(^1\)A simplified version of the preordering component is freely available for research purposes: https://github.com/KantanLabs/KantanPreorder.
Thank you!

Questions?

We thank the EAMT for funding this work!
Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo and Marcello Federico:
*Neural versus Phrase-Based Machine Translation Quality: a Case Study.* In *Proceedings of EMNLP, Austin, USA, 2016.*

Arianna Bisazza and Marcello Federico:

Anita Gojun and Alexander Fraser:
*Determining the placement of German verbs in English–to–German SMT.* In *Proceedings of EACL, Avignon, France, 2012.*

Roger Levy and Galen Andrew:
*Tregex and Tsurgeon: tools for querying and manipulating tree data structures.* In *Proceedings of LREC, Genoa, Italy, 2006.*

Young-Suk Lee, Bing Zhao and Xiaoqiang Luo:

Ole Tange:

Chao Wang, Michael Collins and Philipp Koehn:

Peiyu Wu:
*Word order errors in Simplified Chinese MT.* In *MultiLingual, October/November, 2016.*