Head Finalization: Translation from SVO to SOV

Hideki Isozaki (磯崎 秀樹)

Okayama Prefectural University (岡山県立大学), Japan

December 7, 2012
Long long ago

More than twenty years ago, I had to make a Japanese summary of a chapter of an English book on Artificial Intelligence for a meeting.

I didn’t want to waste time for translation.

I used a commercial RBMT system.

But the result was miserable.

I tried to postedit the output, but it was impossible.

Some sentences lost too much information, and I had to translate it from scratch.

Then I preedited the English source. The result was much better.
Motivation

A few years ago, I was a research scientist of Nippon Telegraph and Telephone Corporation (NTT).

I was developing a cross-lingual medical information retrieval system.

I tried to incorporate an in-house English-to-Japanese HPBMT system into this retrieval system, and found that its output was very poor.

- He took medicine because he became ill.
  was translated as 「彼は薬を飲んだので、病気になった。」 that means
    Because he took medicine, he became ill.

This SMT system tends to SWAP CAUSE AND EFFECT.

*We cannot trust this translator.*
Motivation

Perhaps, our HPBMT system is not the state of the art.

I tried a famous online SMT service.

Even this service made similar mistakes.

Moreover, its JE version translated a Japanese sentence 「メアリはジョンを殺した」 that means “Mary killed John.” as “John killed Mary.”

This service SWAPPED the CRIMINAL AND the VICTIM.

(This problem was fixed recently.)

We cannot trust this service, either.

Thus, wrong word order leads to MISUNDERSTANDING.

I also tried online RBMT services, but they didn’t make such mistakes.
How can we solve the word order problem?

From my experience, it is impossible to postedit translated sentences. We should **preedit** English words.

SMT works very well among European languages.

SMT also works well between Japanese and Korean.

If we can preorder English words into a language whose word order looks like Japanese, SMT will solve other minor problems even if the preordering is not perfect.
My Idea for Preordering English for Japanese

My idea is based on two well known facts.

- Japanese is a **head-final** language.
  
  In Japanese, a modifier (dependent) precedes the modified expression (head). This tendency is called **“head-final”.**
  
  On the other hand, English is a head-initial language.

- We can use an HPSG parser to find heads in an English sentence.

Then, we can implement the following method easily.

1. Parse English sentences with an HPSG parser.
2. If a head precedes its dependent, swap them.
Subject-Object-Verb

Japanese is also called “SOV” or Subject-Object-Verb.

As for “he took medicine”, the object “medicine” is a modifier of the verb “took”.

Therefore, the modifier “medicine” must precede “took” in Japanese.

Both Subject and Object are modifiers of Verb, we can swap them.

```
he =topic medicine =obj took
彼 は 薬 を 飲んだ。
```

```
medicine =obj he =topic took
薬 を 彼 は 飲んだ。
```

Hideki Isozaki (磯崎 秀樹) ()
Head Finalization
December 7, 2012
Head Finalization

Now, we implement the above idea: **Head Finalization**

We use “**Enju**” parser developed at the University of Tokyo.

Enju’s XML output is given in one long line for each sentence.

Here, we pretty-print an example output.

```xml
<sentence id="s0" parse_status="success" fom="25.6314">
  <cons id="c0" cat="S" xcat="" head="c3" sem_head="c3" schema="subj_head">
    <cons id="c1" cat="NP" xcat="" head="c2" sem_head="c2" schema="empty_spec_head">
      <cons id="c2" cat="NX" xcat="" head="t0" sem_head="t0">
        <tok id="t0" cat="N" pos="NNP" base="john" lexentry="[D&lt;N.3sg&gt;]" pred="noun_arg0">
          John
        </tok>
      </cons>
    </cons>
  </cons>:.
</sentence>
```

By focusing on “head” attributes, we can draw the following tree. Thick lines indicate HEADS. Thin lines indicate DEPENDENTS.

We examine this tree in a top-down manner. First, c0’s children c1 and c3 follow the head-final word order. Second, c3’s children c4 and c11 violates the head-final word order. Therefore, we swap c4 and c11 to obtain the head-final word order.
Then, we get this tree.

In the same way, we reorder all head-initial subtrees.
Finally, we get this tree.

We can translate this result (HFE) monotonically into Japanese.

<table>
<thead>
<tr>
<th>English</th>
<th>Japanese</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>ジョン [は]</td>
</tr>
<tr>
<td>Mary</td>
<td>メアリ [が]</td>
</tr>
<tr>
<td>his</td>
<td>彼 の</td>
</tr>
<tr>
<td>wallet</td>
<td>財布 [を]</td>
</tr>
<tr>
<td>lost</td>
<td>なくした</td>
</tr>
<tr>
<td>because</td>
<td>ので</td>
</tr>
<tr>
<td>the</td>
<td>警察</td>
</tr>
<tr>
<td>police</td>
<td>に 行った</td>
</tr>
<tr>
<td>to</td>
<td></td>
</tr>
<tr>
<td>went</td>
<td></td>
</tr>
</tbody>
</table>

δ ϣ Θ [κ] ω Λ [η]  Cosby
In Japanese, we use case markers such as: “は” (topic), “が” (subject), “を” (object), “に” (dative), “の” (genitive, ’s), etc.


English pronoun “his” implicitly has “の”.

English preposition “to” corresponds to “に”.

There is no English words for “は”, “が”, and “を”.

Therefore, we introduce “seed words” to generate these case-markers.
We treat Enju’s arg1 attribute as subject, and arg2 attribute as object.

We introduce seed words “_va1” for arg1 and “_va2” for arg2.

Subjects in the main clause often have topic marker “は”.

But it is very difficult to write down rules to use “は” and “が” properly.

Therefore, we simply replace “_va1” in the main clause with “_va0” and rely on SMT for their proper usage.

John _va0 Mary _va1 his wallet _va2 lost because the police to went
care-no saifu wo nakushita node keisatus ni itta
Coordination Exception

According to Enju’s output, the head of “A and B” is “A”.
If we strictly follow Head Finalization, it becomes “B and A”.
It is logically equivalent, but sometimes the order matters.
Therefore, we do not swap coordination.
This is “Coordination Exception”.

How can we evaluate the effectiveness of Head Finalization?

We use “Kendall’s $\tau$”, a rank correlation coefficient, to measure the similarity of word order between Head Finalized English (HFE) and Japanese.

In order to get $\tau$, we used GIZA++’s alignment file en-ja.A3.final that looks like

John hit a ball.
NULL ({3}) jon ({1}) wa ({}) bohru ({4}) wo ({}) utta ({2}) . ({5})

$$\tau = \frac{\text{# of concordant pairs}}{\text{# of all pairs}} \times 2 - 1$$

$$\tau = \frac{5}{4C_2} \times 2 - 1 = 0.667$$
Distribution of $\tau$ between English and Japanese

We used 1.8 million sentence pairs of NTCIR-7 PATMT.

$\tau$ of Original English

Average of $\tau$: 0.434
Percentage of sentences with $\tau \geq 0.8$: 10.1%

$\tau$ of Head Finalized English

Average of $\tau$: 0.746
Percentage of sentences with $\tau \geq 0.8$: 53.7%
Causes of Low $\tau$ Sentences

- Inexact translation. For example, a Japanese reference sentence for “I bought the cake.” is something like “The cake I bought.”
- Mistakes in Enju’s tagging or parsing.
- Mistakes/Ambiguity in GIZA++’s alignment.

Hideki Isozaki et al.: Head Finalization: A Simple Reordering Rule for SOV Languages, WMT-2010, (W10-1736)
Comparison with Other Methods

We used not only standard BLEU and WER, but also ROUGE-L and IMPACT for this evaluation because Echizenya et al. 2009 showed that ROUGE-L and IMPACT are highly correlated to human evaluation in JE patent translation.

<table>
<thead>
<tr>
<th>Method</th>
<th>dl/mcs</th>
<th>BLEU</th>
<th>ROUGE-L</th>
<th>IMPACT</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>3</td>
<td>0.3361</td>
<td>0.5062</td>
<td>0.4735</td>
<td>0.6354</td>
</tr>
<tr>
<td>Moses PBMT baseline</td>
<td>∞</td>
<td>0.3063</td>
<td>0.4019</td>
<td>0.4022</td>
<td>0.7590</td>
</tr>
<tr>
<td>Moses tree-to-string</td>
<td>20</td>
<td>0.2421</td>
<td>0.3896</td>
<td>0.3926</td>
<td>0.7481</td>
</tr>
<tr>
<td>Moses tree-to-string</td>
<td>∞</td>
<td>0.2450</td>
<td>0.3886</td>
<td>0.3892</td>
<td>0.7770</td>
</tr>
<tr>
<td>Our impl. of Xu et al. ’09</td>
<td>3</td>
<td>0.2554</td>
<td>0.4052</td>
<td>0.4034</td>
<td>0.7438</td>
</tr>
</tbody>
</table>

Head Finalization

References


It is an extension of the WMT-2010 paper.

Head Finalization: A Simple Reordering Rule for SOV Languages, WMT-2010 (W10-1736).
Head Finalization outperformed RBMT

In NTCIR-9 PatentMT task, nine teams participated in EJ subtask. The organizers compared them with two baseline systems, three commercial RBMT systems, and one online translator.

NTT-UT system based on Head Finalization outperformed all RBMTs.

<table>
<thead>
<tr>
<th>system</th>
<th>type</th>
<th>adeq</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTT-UT (RBMT6)</td>
<td>SMT</td>
<td>3.670</td>
</tr>
<tr>
<td>JAPIO (RBMT4)</td>
<td>RBMT</td>
<td>3.507</td>
</tr>
<tr>
<td>(RBMT5)</td>
<td>RBMT</td>
<td>3.463</td>
</tr>
<tr>
<td>(ONLINE)</td>
<td>SMT</td>
<td>3.253</td>
</tr>
<tr>
<td>(Moses HPBMT baseline)</td>
<td>SMT</td>
<td>2.840</td>
</tr>
<tr>
<td>Tottori Univ. (Moses PBMT baseline)</td>
<td>HYBRID</td>
<td>2.667</td>
</tr>
<tr>
<td>POSTECH</td>
<td>SMT</td>
<td>2.603</td>
</tr>
<tr>
<td>Fujitsu R&amp;D Center</td>
<td>SMT</td>
<td>2.477</td>
</tr>
<tr>
<td>Chinese Academy of Science</td>
<td>SMT</td>
<td>2.353</td>
</tr>
<tr>
<td>Univ. of Tokyo</td>
<td>SMT</td>
<td>2.347</td>
</tr>
<tr>
<td>Kyoto Univ.</td>
<td>SMT</td>
<td>2.320</td>
</tr>
<tr>
<td>Beijing Jiaotong Univ.</td>
<td>SMT</td>
<td>2.193</td>
</tr>
</tbody>
</table>

Hideki Isozaki (磯崎 秀樹) ()  Head Finalization  December 7, 2012  20 / 34
Head Finalization outperformed RBMT

References


RIBES
Rank-based Intuitive Bilingual Evaluation Score
We used Kendall’s $\tau$ for evaluation of preordering.

How about using $\tau$ for evaluation of the translation quality?

Source: 彼は雨に濡れたので風邪をひいた

Reference: he caught a cold because he got soaked in the rain

SMT output: he got soaked in the rain because he caught a cold

We use bigrams to disambiguate ambiguous matching.

$\tau$ of the integer list [5, 6, 7, 8, 9, 10, 4, 0, 1, 2, 3] is $-0.236$. 
RIBES is based on “Normalized Kendall’s Tau (NKT)” $(\tau + 1)/2$.

That is, $NKT = \frac{\# \text{ of concordant pairs}}{\# \text{ of all pairs}}$. (concordant pair ratio)

However, we have to consider unmatched words.

We discount NKT by unigram precision $P$.

$$\text{RIBES} = P^\alpha \times NKT \text{ where } 0 \leq \alpha \leq 1.$$
Meta-evaluation of RIBES (NTCIR-7 JE data)

Meta-evaluation is evaluation of automatic evaluation methods by comparing their scores with human judgement scores.

In terms of Spearman’s $\rho$ with adequacy, RIBES gives the best result.

<table>
<thead>
<tr>
<th>Method</th>
<th>adequacy</th>
<th>fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIBES($\alpha = 0.2$)</td>
<td>0.947</td>
<td>0.879</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>0.903</td>
<td>0.889</td>
</tr>
<tr>
<td>IMPACT</td>
<td>0.826</td>
<td>0.751</td>
</tr>
<tr>
<td>METEOR</td>
<td>0.490</td>
<td>0.508</td>
</tr>
<tr>
<td>BLEU</td>
<td>0.515</td>
<td>0.500</td>
</tr>
</tbody>
</table>


Why RIBES is better than BLEU

RBMT tends to use **synonymous** expressions.

BLEU heavily **penalizes synonymous** expressions and doesn’t pay much attention to **global word order**. (single reference)

RIBES heavily **penalizes global word order mistakes** and doesn’t **penalize synonymous** expressions very much.

<table>
<thead>
<tr>
<th></th>
<th>adeq</th>
<th>BLEU</th>
<th>RIBES</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>彼は雨に濡れたので風邪を引いた。</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ref</td>
<td>He caught a cold because he got soaked in the rain.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBMT</td>
<td>He caught a cold because he had gotten wet in the rain.</td>
<td>OK</td>
<td>0.53</td>
</tr>
<tr>
<td>SMT</td>
<td>He got soaked in the rain because he caught a cold.</td>
<td>NG</td>
<td>0.74</td>
</tr>
</tbody>
</table>

**BLEU disagrees with adequacy.**
The meta-evaluation at NTCIR-9 showed that BLEU and NIST are not reliable automatic evaluation metrics for JE and EJ.

<table>
<thead>
<tr>
<th>Method</th>
<th>JE</th>
<th>EJ</th>
<th>CE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>-0.042</td>
<td>-0.029</td>
<td>0.931</td>
</tr>
<tr>
<td>NIST</td>
<td>-0.114</td>
<td>-0.074</td>
<td>0.911</td>
</tr>
<tr>
<td>RIBES</td>
<td>0.632</td>
<td>0.716</td>
<td>0.949</td>
</tr>
</tbody>
</table>

(single reference)

NTT released a Python implementation of RIBES.

In this release, (Strict) Brevity Penalty (BP) was introduced in order to penalize too short output.

Released RIBES = $P^\alpha \times BP^\beta \times NKT \ (0 \leq \beta \leq 1)$

In addition, the bigram restriction in evaluation word alignment was removed.
Language Dependence

Head Finalization worked well for English-to-Japanese translation.

But it has a problem: language dependence.

- Do we have to build HPSG parsers for other languages?
- How about the opposite direction: Japanese-to-English?

Simple “Head Initialization” will not yield good English sentences because English is not a strictly head-intial language.

Head Finalization is already extended to other language pairs.
Chinese-to-Japanese Translation

Han Dan et al. applied Head Finalization to Chinese-to-Japanese Translation.

They used Kun Yu’s Chinese Enju and CWMT (China Workshop on Machine Translation) corpus.

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>RIBES</th>
<th>TER</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CWMT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moses baseline</td>
<td>16.74</td>
<td>71.24</td>
<td>70.86</td>
<td>77.45</td>
</tr>
<tr>
<td>HFC</td>
<td>19.94</td>
<td>73.49</td>
<td>65.19</td>
<td>71.39</td>
</tr>
<tr>
<td>refined HFC</td>
<td><strong>20.79</strong></td>
<td><strong>75.09</strong></td>
<td><strong>64.91</strong></td>
<td><strong>70.39</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CWMT extended</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moses baseline</td>
<td>20.70</td>
<td>74.21</td>
<td>66.10</td>
<td>72.36</td>
</tr>
<tr>
<td>HFC</td>
<td>23.17</td>
<td>75.37</td>
<td>61.38</td>
<td>67.74</td>
</tr>
<tr>
<td>refined HFC</td>
<td><strong>24.14</strong></td>
<td><strong>77.17</strong></td>
<td><strong>59.67</strong></td>
<td><strong>65.31</strong></td>
</tr>
</tbody>
</table>

Japanese-to-English Translation

Katsuhito Sudoh et al. used Head Finalized English (HFE) as a midway point for **Japanese-to-English** Translation.

En-to-Ja:  \[\text{English} \xrightarrow{\text{preordering}} \text{HFE} \xrightarrow{\text{almost monotonic}} \text{Japanese}\]

Ja-to-En:  \[\text{English} \xleftarrow{\text{postordering}} \text{HFE} \xleftarrow{\text{almost monotonic}} \text{Japanese}\]

They used PBMT for both Ja-to-HFE and HFE-to-En.

<table>
<thead>
<tr>
<th>Ja-to-En</th>
<th>BLEU</th>
<th>seconds/sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phrase-based</td>
<td>0.2806</td>
<td>3.532</td>
</tr>
<tr>
<td>Hierarchical Phrase-based</td>
<td>0.2887</td>
<td>7.693</td>
</tr>
<tr>
<td>string-to-tree Syntax-based</td>
<td>0.2686</td>
<td>12.975</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>0.2963</strong></td>
<td>5.462</td>
</tr>
</tbody>
</table>

Isao Goto et al. improved Sudoh’s post-ordering method. They built an **HFE parser** by using the training data of (HFE, swap/straight-labeled Enju Tree) pairs. This improved the post-ordering performance drastically.

<table>
<thead>
<tr>
<th></th>
<th>NTCIR-9</th>
<th>NTCIR-8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RIBES</td>
<td>BLEU</td>
</tr>
<tr>
<td>Proposed</td>
<td>94.66</td>
<td>80.02</td>
</tr>
<tr>
<td>PBMT Post-ordering</td>
<td>77.34</td>
<td>62.24</td>
</tr>
<tr>
<td>HPBMT Post-ordering</td>
<td>77.99</td>
<td>53.62</td>
</tr>
</tbody>
</table>

Acknowledgements

The author would like to thank Prof. Yusuke Miyao, who answered my questions on Enju and sometimes improved the Enju system for my requests.

The author also thanks members of NTT Communication Science Laboratories for supporting my research.
enjutree package is available for \LaTeX TikZ

\usepackage{enjutree}
\begin{document}
\begin{enjutree}{}
\begin{sentence id="s0" parse_status="success" fom="25.6314">
\begin{cons id="c0" cat="S" xcat="" head="c3" sem_head="c3" schema="subj_head">
\end{cons}
\end{sentence}
\end{enjutree}
\end{document}