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?

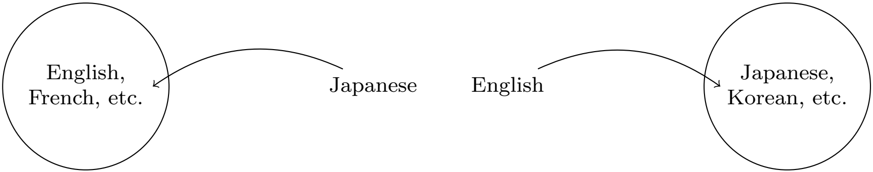
From my experience, it is impossible to postedit translated sentences.

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-finial** 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。
S	O	V

medicine =obj	he =topic	took
薬 を	彼 は	飲んだ。
O	S	V

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.

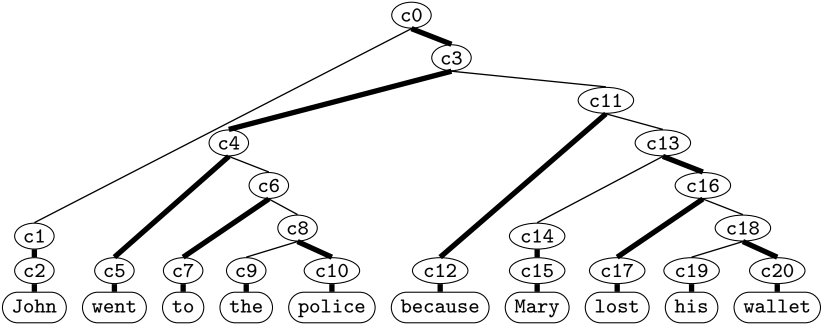
```
<sentence id="s0" parse_status="success" fom="25.6314">
 <cons id="c0" cat="S" xcat="" head="c3" sem_head="c3" schema="subj_head">
  <cons id="c1" cat="NP" xcat="" head="c2" sem_head="c2" schema="empty_spec_head">
   <cons id="c2" cat="NX" xcat="" head="t0" sem_head="t0">
    <tok id="t0" cat="N" pos="NNP" base="john" lexentry="[D&lt;N.3sg&gt;]" pred="no
   </cons>
  </cons>
    :
 </cons>.
</sentence>
```

Yusuke Miyao and Jun'ichi Tsujii: Feature Forest Models for Probabilistic HPSG Parsing, Computational Linguistics, Vol.34, No.1, pp.81-88, 2008. (J08-1002)

Head Finalization

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.

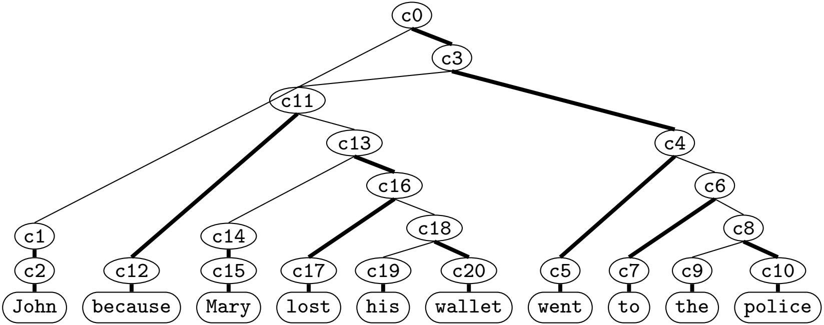
First, `c0`’s children `c1` and `c3` follow the head-final word order.

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.

Head Finalization

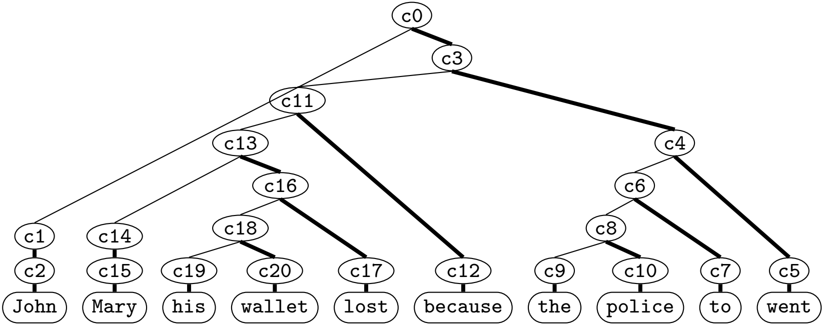
Then, we get this tree.



In the same way, we reorder all head-initial subtrees.

Head Finalization

Finally, we get this tree.



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

```
John        Mary        his      wallet      lost       because the police   to went
jon   [wa] meari [ga] kare no saifu [wo] nakushita node        keisatus ni itta
ジョン [は] メアリ [が]  彼   の  財布 [を]  なくした   ので          警察     に 行った
```

Seed Words for Case Markers

In Japanese, we use case markers such as: “は” (wa) (topic), “が” (ga) (subject), “を” (wo) (object), “に” (ni) (dative), “の” (no) (genitive, `'s`), etc.

```
John        Mary        his       wallet       lost        because the police    to went
jon   [wa]  meari [ga]  kare no   saifu [wo]   nakushita   node        keisatus  ni itta
ジョン [は]  メアリ [が]  彼   の    財布 [を]     なくした      ので          警察      に 行った
```

English pronoun “`his`” implicitly has “の” (no).

English preposition “`to`” corresponds to “に” (ni).

There is no English words for “は” (wa), “が” (ga), and “を” (wo).

Therefore, we introduce “seed words” to generate these case-markers.

Seed Words for Case Markers

We treat Enju's `arg1` attribute as subject, and `arg2` attribute as object.

```
<tok id="t7" cat="V" pos="VBD" base="lose" lexentry="[NP.nom&lt;V.bse&gt;NP.acc]-pa
     pred="verb_arg12" tense="past" aspect="none" type="none" voice="active" aux="m
     arg1="c14" arg2="c18">lost</tok>
```

We introduce seed words "`_va1`" for arg1 and "`_va2`" for arg2.

Subjects in the main clause often have topic marker "は" (wa).

But it is very difficult to write down rules to use "は" (wa) and "が" (ga) 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
jon   wa  meari ga     kare-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**”.

Evaluation of Head Finalization

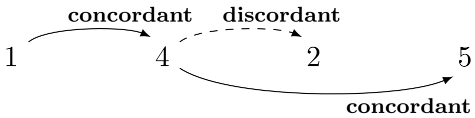
How can we evaluate the effectiveness of Head Finalization?

We use "**Kendall's** $\boldsymbol{\tau}$", a rank correlation coefficient, to measure the similarity of word order between Head Finalized English (HFE) and Japanese.

In otder to get τ, 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})
```

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

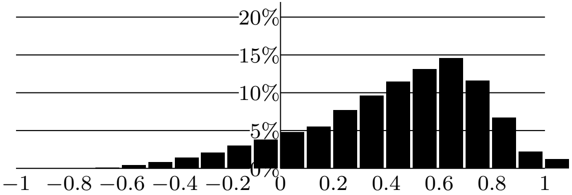


$$\tau = \frac{5}{{}_4C_2} \times 2 - 1 = 0.667$$

Distribution of τ between English and Japanese

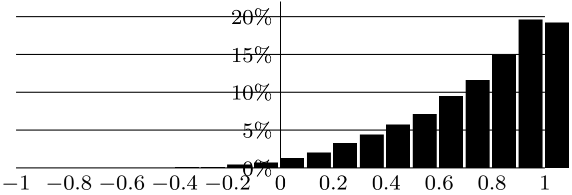
We used 1.8 million sentence pairs of NTCIR-7 PATMT.

τ of Original English



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

τ of Head Finalized English



Average of τ: 0.746
Percentage of sentences with $\tau \geq 0.8$: 53.7%

Causes of Low τ 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.

	dl/mcs	BLEU	ROUGE-L	IMPACT	WER
Proposed	3	**0.3361**	**0.5062**	**0.4735**	**0.6354**
Moses PBMT baseline	∞	0.3063	0.4019	0.4022	0.7590
Moses tree-to-string	20	0.2421	0.3896	0.3926	0.7481
Moses tree-to-string	∞	0.2450	0.3886	0.3892	0.7770
Our impl. of Xu et al. '09	3	0.2554	0.4052	0.4034	0.7438

Hideki Isozaki et al.: HPSG-based Preprocessing for English-to-Japanese Translation, ACM Transactions on Asian Language Information Processing, Vol.11, Issue 3, Article 8, 16 pages, September 2012. ACM TALIP

Head Finalization

References

Hideki Isozaki et al.: HPSG-based Preprocessing for English-to-Japanese Translation, ACM Transactions on Asian Language Information Processing, Vol.11, Issue 3, Article 8, 16 pages, September 2012. ACM TALIP

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 orgnizers compared them with two baseline systems, three commercial RBMT systems, and one online translator.

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

system	type	adeq
NTT-UT	**SMT**	**3.670**
(RBMT6)	RBMT	3.507
JAPIO	RBMT	3.463
(RBMT4)	RBMT	3.253
(RBMT5)	RBMT	2.840
(ONLINE)	SMT	2.667
(Moses HPBMT baseline)	SMT	2.603
Tottori Univ.	HYBRID	2.600
(Moses PBMT baseline)	SMT	2.477
POSTECH	SMT	2.353
Fujitsu R&D Center	SMT	2.347
Chinese Academy of Science	SMT	2.320
Univ. of Tokyo	SMT	2.193
Kyoto Univ.	SMT	2.180
Beijing Jiaotong Univ.	SMT	1.793

Head Finalization outperformed RBMT

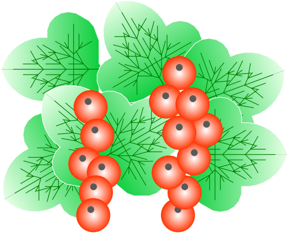
References

Isao Goto et al.: Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop, in Proc. of NTCIR-9, pp.559–578, 2012. NTCIR9-GotoI

Sudoh et al.: NTT-UT Statistical Machine Translation in NTCIR-9 PatentMT, in Proc. of NTCIR-9, pp.585–592, 2012. NTCIR9-SudohK

RIBES

Rank-based Intuitive Bilingual Evaluation Score

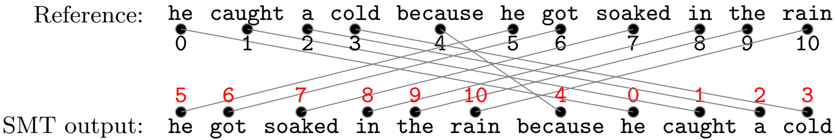


RIBES

We used Kendall's τ for evaluation of preordering.

How about using τ for evaluation of the **translation quality**?

kare wa ame ni nureta node kaze wo hiita
Source: 彼 は 雨 に 濡れた ので 風邪 を ひいた



We use bigrams to disambiguate ambiguous matching.

τ of the integer list [5, 6, 7, 8, 9, 10, 4, 0, 1, 2, 3] is -0.236.

RIBES

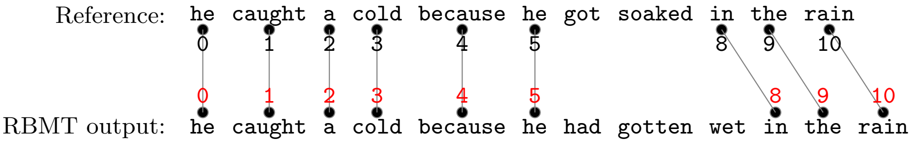
RIBES is based on “Normalized Kendall’s Tau (NKT)” $(\tau + 1)/2$.

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

However, we have to consider unmatched words.

We discount NKT by unigram precision P.

$$\text{RIBES} = P^{\alpha} \times \text{NKT where } 0 \leq \alpha \leq 1.$$



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 ρ with adequacy, RIBES gives the best result.

Method	adequacy	fluency
RIBES($\alpha = 0.2$)	**0.947**	0.879
ROUGE-L	0.903	**0.889**
IMPACT	0.826	0.751
METEOR	0.490	0.508
BLEU	0.515	0.500

(single reference)

Isozaki et al.: Automatic Evaluation of Translation Quality for Distant Language Pairs, EMNLP, pp.944-052, 2010. (D10-1092)

Hirao et al.: RIBES: Automatic Evaluation of Translation Quality based on Rank Correlation (in Japanese), Proc. of Annual Conference on Natural Language Processing, pp.1115–1118, 2011.

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.

		adeq	BLEU	RIBES
source	彼は雨に濡れたので風邪を引いた。			
Ref	He caught a cold because he got soaked in the rain.			
RBMT	He caught a cold because he had gotten wet in the rain.	OK	0.53	**0.93**
SMT	He got soaked in the rain because he caught a cold.	NG	0.74	**0.38**

BLEU disagrees with adequacy.

Meta-evaluation at NTCIR-9

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

Method	JE	EJ	CE
BLEU	−0.042	−0.029	0.931
NIST	−0.114	−0.074	0.911
RIBES	**0.632**	**0.716**	**0.949**

(single reference)

Isao Goto et al.: Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop, Proc. of NTCIR-9, pp.559–578, 2012. NTCIR9-GotoI

RIBES is available from NTT

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 $= P^{\alpha} \times \mathrm{BP}^{\beta} \times \mathrm{NKT}$ $(0 \leq \beta \leq 1)$

In addition, the bigram restriction in evaluation word alignmnet 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.

	BLEU	RIBES	TER	WER
	CWMT			
Moses baseline	16.74	71.24	70.86	77.45
HFC	19.94	73.49	65.19	71.39
refined HFC	**20.79**	**75.09**	**64.91**	**70.39**
	CWMT extended			
Moses baseline	20.70	74.21	66.10	72.36
HFC	23.17	75.37	61.38	67.74
refined HFC	**24.14**	**77.17**	**59.67**	**65.31**

Han Dan et al.: Head Finalization Reordering for Chinese-to-Japanese Machine Translation, In Proc. of SSST-6, Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation, pp.57–66, 2012. (W12-4207)

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: English —preordering→ HFE —almost monotonic→ Japanese

Ja-to-En: English ←**postordering**— HFE ←almost monotonic— Japanese

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

Ja-to-En	BLEU	seconds/sentence
Phrase-based	0.2806	3.532
Hierarchical Phrase-based	0.2887	7.693
string-to-tree Syntax-based	0.2686	12.975
Proposed	**0.2963**	5.462

Katsuhito Sudoh et al.: Post-ordering in Statistical Machine Translation, In Proc. of the 13th Machine Translation Summit, pp.316–323, 2011.

Japanese-to-English Translation

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.

oracle-HFE-to-En	NTCIR-9		NTCIR-8	
	RIBES	BLEU	RIBES	BLEU
Proposed	**94.66**	**80.02**	**94.93**	**79.99**
PBMT Post-ordering	77.34	62.24	78.14	63.14
HPBMT Post-ordering	77.99	53.62	80.85	58.34

Isao Goto et al.: Post-ordering by Parsing for Japanese-English Statistical machine Translation, In Proc. of the 50th Annual Meeting of the Association for Computational Linguistics, pp.311–316, 2012. (P12-2061)

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