HPSG-Based Preprocessing for English-to-Japanese Translation

HIDEKI ISOZAKI, KATSUHITO SUDOH, HAJIME TSUKADA, and KEVIN DUH,
NTT Corporation

Japanese sentences have completely different word orders from corresponding English sentences. Typical phrase-based statistical machine translation (SMT) systems such as Moses search for the best word permutation within a given distance limit (distortion limit). For English-to-Japanese translation, we need a large distance limit to obtain acceptable translations, and the number of translation candidates is extremely large. Therefore, SMT systems often fail to find acceptable translations within a limited time. To solve this problem, some researchers use rule-based preprocessing approaches, which reorder English words just like Japanese by using dozens of rules. Our idea is based on the following two observations: (1) Japanese is a typical head-final language, and (2) we can detect heads of English sentences by a head-driven phrase structure grammar (HPSG) parser. The main contributions of this article are twofold: First, we demonstrate how off-the-shelf, state-of-the-art HPSG parser enables us to write the reordering rules in an abstract level and can easily improve the quality of English-to-Japanese translation. Second, we also show that syntactic heads achieve better results than semantic heads. The proposed method outperforms the best system of NTCIR-7 PATMT EJ task.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms: Algorithms, Experimentation, Languages, Performance

Additional Key Words and Phrases: Machine translation, Japanese, English, SOV, SVO, HPSG

ACM Reference Format:
Isozaki, H., Sudoh, K., Tsukada, H., and Duh K. 2012. HPSG-based preprocessing for English-to-Japanese translation. ACM Trans. Asian Lang. Inform. Process. 11, 3, Article 8 (September 2012), 16 pages.
DOI = 10.1145/2334801.2334802 http://doi.acm.org/10.1145/2334801.2334802

1. INTRODUCTION

Statistical machine translation (SMT) [Koehn 2009] works well for machine translation among languages that have similar word orders. However, SMT does not work well for distant language pairs such as English and Japanese. English is a subject-verb-object (SVO) language and Japanese is a subject-object-verb (SOV) language.

Some existing methods try to solve this word-order problem in *language-independent* ways [Genzel 2010; Li et al. 2007; Quirk et al. 2005; Xia and McCord 2004; Yamada and Knight 2001]. They usually parse input sentences and learn reordering decisions at each node of the parse trees.

Other methods tackle this problem in *language-dependent* ways [Collins et al. 2005; Katz-Brown and Collins 2008; Lee et al. 2010; Nguyen and Shimazu 2006]. If

An earlier version of this article [Isozaki et al. 2010] was presented at *ACL Joint Fifth Workshop on Statistical Machine Translation and Metrics (MATR)*.

Authors' addresses: H. Isozaki, Faculty of Computer Science and Systems Engineering, Okayama Prefectural University, Kuboki-111, Soja-shi, Okayama, 719-1197, Japan; email: isozaki@cse.oka-pu.ac.jp; K. Sudoh, H. Tsukada, and K. Duh, NTT Communication Science Laboratories, NTT Corporation, 2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan.

DOI 10.1145/2334801.2334802 http://doi.acm.org/10.1145/2334801.2334802

we can completely reorder words by preprocessing, we will be able to greatly reduce the computational cost of SMT systems. Recently, Xu et al. [2009] and Hong et al. [2009] proposed rule-based preprocessing methods for SVO-to-SOV translation. These methods parse input sentences by dependency parsers and reorder the words using a set of hand-crafted rules to obtain SOV-like sentences. Hong et al.'s method [2009] is straightforward. They obtain SOV word order by moving verbs and adjectives to the ends of corresponding phrases. Xu et al. [2009] used more complicated rules. Each of their rules is composed of a part-of-speech tag, a dependency label, a rule weight, and a reordering order.

On the other hand, Lee et al. [2010] employed a constituent parser instead of dependency parsers. A constituent parser provides the grammatical function tag of each phrase. By using the function tags, it is easier to specify the context where a rule is applicable.

In this article, we present one general reordering rule: Head Finalization. Our idea is simple: move heads to the end of corresponding phrases or clauses because Japanese is a typical head-final language. That is, a head word usually comes after dependent words. SOV is just one aspect of head-final languages. Head-driven phrase structure grammar (HPSG [Pollard and Sag 1994]) parsers enable us to implement the rule at an abstract level. To implement this idea, we need a parser that outputs "syntactic heads." Enju [Miyao and Jun'ichi Tsujii 2008; Miyao and Tsujii 2005] from the University of Tokyo is such a parser[1]. In this article, we use Enju-2.4.1. We discuss other parsers in Section 5.

There is another kind of head: "semantic heads." Hong et al. [2009] used Stanford parser [de Marneffe et al. 2006], which outputs semantic head-based dependencies; Xu et al. [2009] also used the same semantic dependencies, but they did not mention what parser they used. Enju also outputs semantic heads. Rough definitions of these heads are given as follows.

Syntactic head: A part of a phrase that determines the syntactic role of the phrase.
Semantic head: A part of a phrase that determines the main meaning of the phrase.

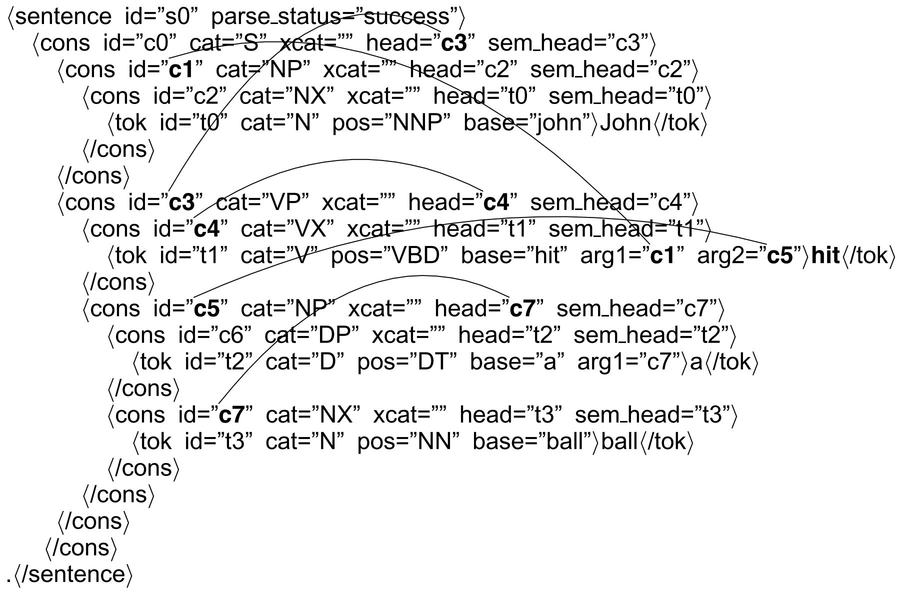
Exact definitions may differ by researchers and parsers, but we do not discuss their definitions here. We simply accept the output of these parsers and evaluate them in terms of English-to-Japanese translation quality.

Our method simply checks whether a tree node is a syntactic head. We do not have to care about what we move or how we move it. The use of syntactic heads and the number of dependents are essential for the simplicity of Head Finalization (See Discussion). Enju outputs binary trees, which leads to simple implementation. If we use an ordinary dependency parser that does not limit the number of dependents, we need a "reverse" operation just like Xu et al. [2009]. The binary tree constraint is known in SMT as 'inversion transduction grammar (ITG) constraints' [Wu 1997].

2. HEAD FINALIZATION

Figure 1 shows Enju's XML output for a simple sentence: "`John hit a ball`." The tag "`<cons>`" indicates a nonterminal node and "`<tok>`" indicates a terminal node or a word (token). Each node has a unique `id`. Its syntactic head is given by the node's "`head`" attribute. For instance, node `c0`'s syntactic head is node `c3`, and `c3` is a `VP`, or verb phrase. Thus, Enju treats not only words but also non-terminal nodes as heads. Its semantic head is given by "`sem_head`" attribute. In this example, syntactic heads and

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/enju



```
⟨sentence id="s0" parse_status="success"⟩
  ⟨cons id="c0" cat="S" xcat="" head="c3" sem_head="c3"⟩
    ⟨cons id="c1" cat="NP" xcat="" head="c2" sem_head="c2"⟩
      ⟨cons id="c2" cat="NX" xcat="" head="t0" sem_head="t0"⟩
        ⟨tok id="t0" cat="N" pos="NNP" base="john"⟩John⟨/tok⟩
      ⟨/cons⟩
    ⟨/cons⟩
    ⟨cons id="c3" cat="VP" xcat="" head="c4" sem_head="c4"⟩
      ⟨cons id="c4" cat="VX" xcat="" head="t1" sem_head="t1"⟩
        ⟨tok id="t1" cat="V" pos="VBD" base="hit" arg1="c1" arg2="c5"⟩hit⟨/tok⟩
      ⟨/cons⟩
      ⟨cons id="c5" cat="NP" xcat="" head="c7" sem_head="c7"⟩
        ⟨cons id="c6" cat="DP" xcat="" head="t2" sem_head="t2"⟩
          ⟨tok id="t2" cat="D" pos="DT" base="a" arg1="c7"⟩a⟨/tok⟩
        ⟨/cons⟩
        ⟨cons id="c7" cat="NX" xcat="" head="t3" sem_head="t3"⟩
          ⟨tok id="t3" cat="N" pos="NN" base="ball"⟩ball⟨/tok⟩
        ⟨/cons⟩
      ⟨/cons⟩
    ⟨/cons⟩
  ⟨/cons⟩
.⟨/sentence⟩
```

Fig. 1. Enju's XML output. (Some attributes are omitted for readability.)

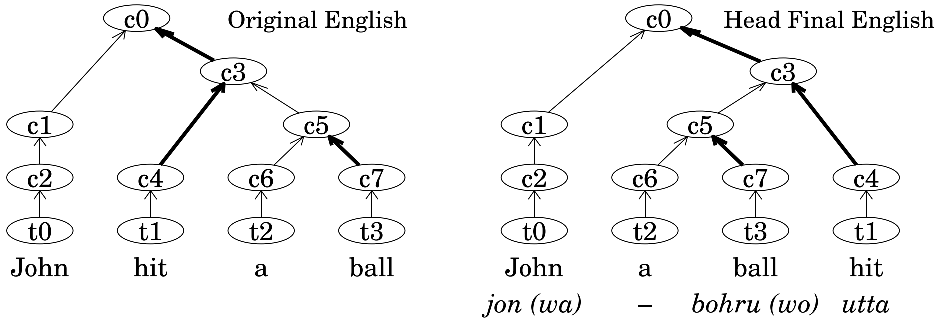


Fig. 2. Head Finalization of a simple sentence. (Thick lines indicate edges from heads.)

semantic heads agree, but they sometimes disagree. We will discuss this disagreement in Section 4.2.

Enju outputs at most two child nodes for each node. One child is a syntactic head, and the other is a dependent. c3's syntactic head is c4, which is VX, or a fragment of a verb phrase. c4's syntactic head is t1 or "hit," which is VBD or a past-tense verb. The left picture of Figure 2 shows the parse tree graphically. Here, thick lines indicate edges from syntactic heads.

Our Head Finalization rule simply swaps two children of each node when the head child appears before the dependent child. In the left picture of Figure 2, c3 has two children c4 and c5. Here, c3's head c4 appears before c5, so c4 and c5 are swapped. The right picture shows the swapped result. Then we obtain "John a ball hit," which has the same word order as its Japanese translation "jon wa bohru wo utta" except for the English determiner "a" and Japanese case markers "wa" and "wo". Our preliminary experiments showed that Head Final English (HFE) looks good as a first approximation of Japanese word order. In addition, we can make it better by introducing some heuristic rules.

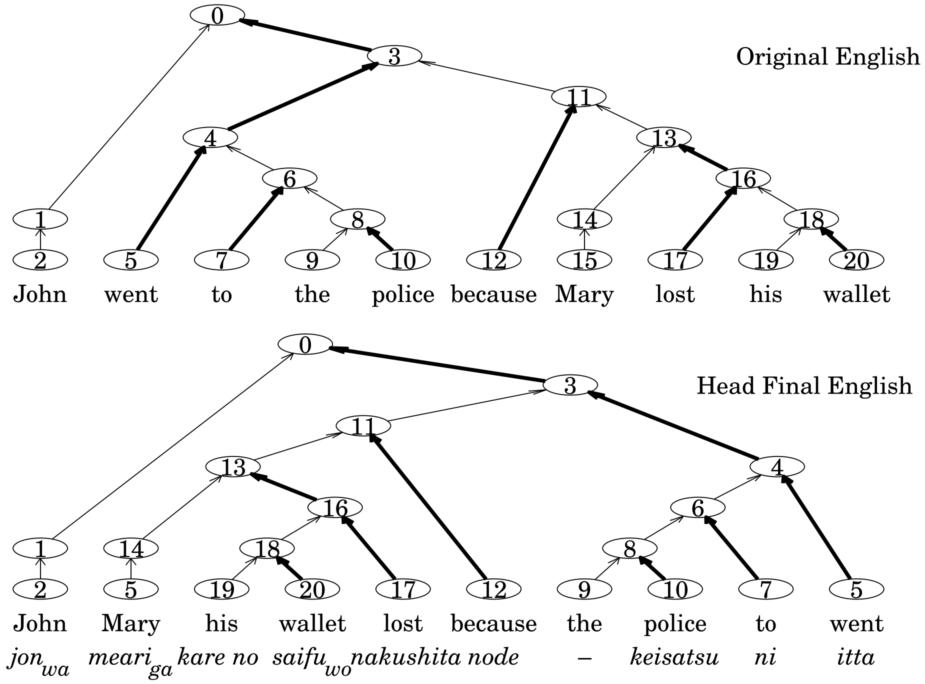

Original English
0
3
11
4
13
6
16
1
8
14
18
2 5 7 9 10 12 15 17 19 20
John went to the police because Mary lost his wallet
Head Final English
0
3
11
4
13
16
6
1
14
18
8
2 5 19 20 17 12 9 10 7 5
John Mary his wallet lost because the police to went
jon wa meari ga kare no saifu wo nakushita node – keisatsu ni itta

Fig. 3. Head-Finalizing a complex sentence. (Thick lines indicate edges from heads.)

2.1. Processing Words Without Counterparts

We have to add Japanese case markers “wa” (topic marker) or “ga” (nominative case marker) for “John” and “wo” (objective case marker) for “ball” to obtain an acceptable Japanese sentence. SMT is not good at generating appropriate case markers from English. Case marker generation has been tackled by a few research groups [Hong et al. 2009; Toutanova and Suzuki 2007].

Here, we use Enju's output to generate “seeds for case markers.” As Figure 1 shows, the verb “hit” has `arg1 = "c1"` and `arg2 = "c5"`. This indicates that `c1 ("John")` is the first argument (subject) of “hit” and `c5 ("a ball")` is the second argument (object) of “hit”. We add seed words “_va1” after `arg1` and “_va2” after `arg2`. Then, we obtain “`John _va1 a ball _va2 hit`.” We introduced the idea of case marker seed words independently but found that this is very similar to Hong et al. [2009] for Korean. Lee et al. [2010] also used a similar method.

Figure 3 shows Enju's parse tree for a more complicated sentence “`John went to the police because Mary lost his wallet.`” For brevity, we hide terminal nodes, and we removed non-terminal nodes' prefix `"c"`.

We improved the introduction of case marker seed words in the following way.

— For Figure 3, Japanese speakers will use “jon wa meari ga” instead of “jon ga meari ga” because the latter sounds like a correction of the subject. To help this, we simply rewrite “_va1” of the main clause as “_va0” in each sentence to distinguish between these two case markers. The distinction of case markers such as “ga” and “wa” is a delicate problem, and we rely on SMT's statistical analysis for the determination of case markers.

— By examining the output, we found that some words have two seed words. For example, “toy” of “`He bought a toy that was popular in Japan`” has two seed words "_va1" and "_va2" because the toy is a subject in the relative clause and an object in the main clause. By considering Japanese translation, we decided to discard seed words from the relative clause.

- For *passive* sentences such as "John was hit by Bob," Enju treats "Bob" as the first argument and "John" as the second argument of "hit" because semantically Bob hit John. However, passive English sentences are often translated into passive Japanese sentences and this argument swap for passive verbs may confuse the translation system in such cases. Therefore, we undo Enju's argument swap to achieve straightforward Japanese translations. That is, we treat the subject of the passive English sentence as the first argument of the verb.
  Sometimes, passive sentences are translated into active sentences, but we have to consider the focus of the story in the target language (Japanese). If we translate a passive sentence into an active sentence, its subject or the focus of the story changes. The focus is usually used as a default subject in Japanese. Therefore, we should not change subjects without considering the context.

On the other hand, English determiners ("a," "an," and "the") are not translated into any Japanese words in most cases. Sometimes, however, they are explicitly translated ("the" ⇒ "kono") or implicitly affect the translation ("a few" vs. "few"), but we simply remove these determiners here.

Conventional rule-based machine translation (RBMT) systems swap X and Y of "X because Y" and move verbs to the end of each clause. Then we obtain "Mary his wallet lost because John the police to went." Its word-to-word translation is a natural Japanese sentence: "meari (ga) kare no saifu (wo) nakushita node jon (wa) keisatsu ni itta."

Our Head Finalization with case marker seed words yields a slightly different word order "John _va0 Mary _va1 his wallet _va2 lost because the police to went." Its word-to-word translation is "jon wa meari ga kare no saifu wo nakushita node keisatsu ni itta." This is also an acceptable Japanese sentence. This difference comes from the syntactic role of "because." In our method, Enju states that "because" is a dependent of "went," whereas RBMT systems treat "because" as a clause conjunction.

2.2. Stopping Excessive Reordering

Although Head Finalization works very well for ordinary sentences, some expressions are excessively reordered. From preliminary experiments, we found that "A and B" was reordered as "B and A." Logically, they are equivalent, but sometimes their order matters. Therefore, we decided to exclude coordination nodes (cat = "COOD" or xcat = "COOD") from this reordering. We call this the *coordination exception rule* and implement it by treating the latter part B as the head of the coordination node.

Technical documents have more elements that cause excessive reordering problems.

(1) Colons and semicolons. Colons and semicolons are used to concatenate (sub)sentences, and they cause excessive reordering. We solve this by splitting these sentences at colons and semicolons. After the Head Finalization, these (sub)sentences are concatenated again to obtain the entire Head Final English sentence.

(2) Patent documents often have expressions such as "motor 1" which means "first motor." Here, "motor" is the head and "1" is its dependent. Head Finalization reorders it as "1 motor" and we can translate it monotonically as "daiichi mohtah (first motor)." However, we cannot distinguish "1 motor" from "one motor." In addition, "motor 1" is monotonically translated as "mohtah 1" in most patent documents.

(3) Parentheses: For example, "`(1)`" is reordered as "`) 1 (`." We can expect distance-based SMT systems with a small distortion limit to learn how to correct such local reordering mistakes, but this violates Japanese word order.
In addition, Enju's part-of-speech tagger "`stepp`" trained with Penn's *Wall Street Journal* data (just like other taggers) treats brackets "`[`" and "`]`" as unknown words and sometimes assigns *verb tags* to them. This causes severe parsing mistakes. (This bracket problem was corrected in Enju 2.4.2.)

Therefore, we introduce a program "`repos`" that rewrites `stepp`'s output as follows.

- Rewrite POS tag `CD` to `NN`. This keeps "`motor 1`".
- Rechunk short parentheses such as "`(1)`" and assign `NN` to it. This keeps "`(1)`".
- Assign `-LRB-` to `"["` and `-RRB-` to `"]"`.

3. EXPERIMENTS

To show how closely our Head Finalization makes English follow Japanese word order, we measured Kendall's τ [Kendall 1975], a rank correlation coefficient. We also measured BLEU [Papineni et al. 2002] and other automatic evaluation scores to show that Head Finalization can actually improve the translation quality.

We used NTCIR7 PATMT's Patent corpus [Fujii et al. 2008]. Its training corpus has 1.8 million sentence pairs. We used MeCab[2] to segment Japanese sentences into words.

3.1. Rough Evaluation of Preprocessing

First, we examined rank correlation between HFE sentences produced by the Head Finalization rule and Japanese reference sentences. Since we do not have handcrafted word alignment data for an English-to-Japanese bilingual corpus, we used GIZA++ [Och and Ney 2003] to automatically align words.

On the basis of this automatic word alignment, we measured Kendall's τ for the word order between HFE sentences and Japanese sentences. Kendall's τ is a kind of rank correlation measure defined as follows. Suppose a list of integers such as L = [2, 1, 3, 4]. The number of all integer pairs in this list is ${}_4C_2 = 6$. The number of increasing pairs is five: (2, 3), (2, 4), (1, 3), (1, 4), and (3, 4). Kendall's τ is defined by

$$\tau = \frac{\#\text{increasing pairs}}{\#\text{all pairs}} \times 2 - 1.$$

In this case, we obtain $\tau = 5/6 \times 2 - 1 = 0.667$. For each sentence in the training data, we calculate τ on the basis of a GIZA++'s alignment file "`en-ja.A3.final`", which looks like this:

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

Numbers in `({ })` indicate corresponding English words. "`NULL`" indicates English words without Japanese counterparts. From this alignment, we obtain an integer list [1, 4, 2, 5]. Then, τ is $5/{}_4C_2 \times 2 - 1 = 0.667$. For HFE in Figure 2, we will achieve the following alignment.

```
John _va0 ball _va2 hit .
NULL ({}) jon ({1}) wa ({2}) bohru ({3}) wo ({4}) utta ({5}) . ({6})
```

[2] http://mecab.sourceforge.net/

Then, we obtain [1, 2, 3, 4, 5, 6] and $\tau = 1.0$. Sometimes, one Japanese word corresponds to an English phrase:

```
John went to Costa Rica .
NULL ({}) jon ({1}) wa ({}) kosutarika ({4 5}) ni ({3}) itta ({2}) . ({6})
```

We obtain [1, 4, 5, 3, 2, 6] from this alignment. In the following case, two occurrences of "`of`" correspond to one Japanese word "*no*."

```
rate of change of speed
NULL ({}) sokudo ({5}) henka ({3}) no ({2 4}) wariai ({1})
```

We excluded the discontiguously aligned words `(2 4)` from the calculation of τ. We use only [5, 3, 1] and obtain $\tau = -1.0$. Because of this exclusion, the best value $\tau = 1.0$ does not mean that we obtained the perfect word ordering, but low τ values imply failures. Thus, even this rough calculation of τ sheds light on the weak points of Head Finalization.

3.2. Quality of Translation

Note that the rank correlation does not directly measure the quality of translation. For translation, we use Moses [Koehn 2010] with distance-based reordering. Usually, the quality of translation is measured by BLEU [Papineni et al. 2002]. However, Echizen-ya et al. [2009] showed that ROUGE-L [Lin and Och 2004] and their IMPACT method are highly correlated to human evaluation in evaluating J-E patent translation. We also used other evaluation measures: word error rate (WER) [Su et al. 1992], position-independent error rate (PER), and translation error (or edit) rate (TER) [Snover et al. 2006].

We used the development set (915 sentences) in the NTCIR7 PATMT data as well as the formal run test set (1,381 sentences). PATMT participants used different methods such as phrase-based SMT [Kumai et al. 2008], *hierarchical phrase*-based SMT [Watanabe et al. 2008], RBMT [Izuha et al. 2008], and example-based machine translation (EBMT) [Nakazawa and Kurohashi 2008]. However, the organizers' Moses-based baseline system denoted as "Moses*" obtained the best BLEU score.

4. RESULTS

First, we show τ values to evaluate word order, and then we show BLEU and other automatic evaluation scores.

4.1. Quality of Preprocessing

The left subfigure of Figure 4 shows the distribution of τ for the 1.8 million training sentences. The average τ ($\overline{\tau}$) over all training sentences was $\overline{\tau} = 0.434$ and the percentage of sentences with $\tau \geq 0.8$ was 10.1% when we used the original English sentences.

The right subfigure shows how Head Finalization improved the distribution. Head Finalization with "`repos`" improved the former to 0.746 and the latter to 53.7%. The percentage of sentences with $\tau = 1.0$ was 19.2%. Without `repos`, $\overline{\tau}$ was slightly degraded to 0.729. The ratio of high $\tau (\geq 0.8)$ sentences was 51.4% and that of sentences with $\tau = 1.0$ was 7.0%. Thin bars in the right subfigure show the distribution of τ of HFE without `repos`. `repos` increases the ratio of $\tau = 1.0$ and reduces the ratio of smaller τ.

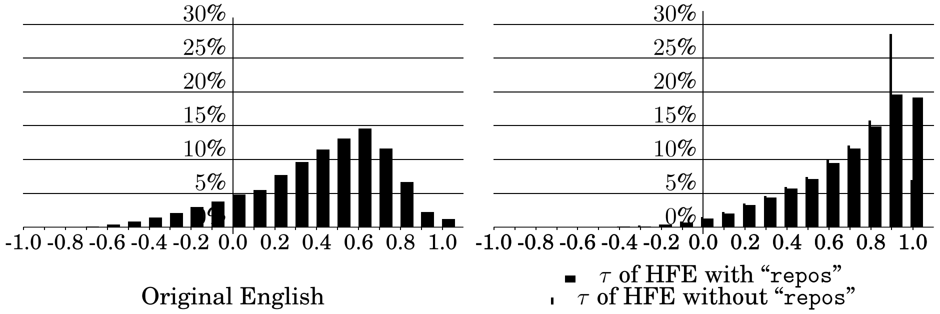


Fig. 4. Distribution of τ.

In a preliminary experiment, we examined low τ sentences of our method and found the following reasons for low τ values.

- The sentence pair is not an exact one-to-one translation. A Japanese reference sentence for "`I bought the cake.`" can be something like "`The cake I bought.`" or "`The person who bought the cake is me.`"
- Mistakes in Enju's tagging or parsing. Just like other parsers, Enju tends to make mistakes when a sentence has a comma or '`and`.' When we used Enju-2.3.1, we often encountered severe mistakes caused by POS tag errors such as VBN/VBD (e.g., killed) and VBZ/NNS (e.g., reports). Here, we used Enju-2.4.1's `-A` option that takes such ambiguous POS tag cases into account.
- Mistakes/Ambiguity of GIZA++ automatic word alignment. As we described above, discontiguously aligned words are removed from calculation of τ. Thus, small reordering mistakes in other words are emphasized.

4.2. Syntactic Heads vs. Semantic Heads

As we described above, Stanford parser outputs "semantic heads" instead of "syntactic heads." The examples in Xu et al.'s article [2009] imply that semantic heads are not as good as Enju's syntactic heads at achieving Japanese-like word order.

Their Figure 3 shows dependency of "`John can hit the ball`." The root of this sentence is not "`can`" but "`hit`." Therefore, Head Finalization based on semantic heads does not move "`can`" to the end of the sentence, and they need an "aux" rule to move "`can`" after "`hit`."

Their Figure 4 shows dependency of "`Living is exciting because we don't know what the future has`." Here, "`because`" is just a modifier of "`know`" and Head Finalization does not move "`because`" to the end of this "`because`" clause. Therefore, they proposed a new rule in their footnote.

To justify our belief that semantic heads are not as good as syntactic heads at achieving Japanese-like word order, we implemented two systems. The first system is our implementation of Xu et al.'s method [2009] with Stanford parser 1.6.2. We encountered and solved the following problems. (We describe their system in Section 5.)

- This version of Stanford parser sometimes output cyclic dependency relationship, and we could not reach some words from the root word of the sentence.
- Xu et al. [2009] disallowed any movement across punctuation and conjunctions. In the case of "John saw Bob and Mary," we want to move "Bob and Mary" together. However, it is not clear how they treated such cases.

The second system is a *Semantic Head Finalization* program, which simply uses Enju's semantic heads (indicated by "`sem_head`" attribute) instead of syntactic heads.

Table I. Usage of Semantic Heads

Method	English	Xu-like	Semantic HFE	**HFE**	HFE-va
Averaged τ	0.434	0.624	0.716	**0.746**	0.721
$\tau \geq 0.8$	10.1%	30.6%	49.4%	**53.7%**	50.3%

Table I compares these methods with (syntactic) HFE. According to these results, syntactic heads achieve better results than semantic heads. We also show HFE-va or HFE without case marker seed words. This also degrades the performance.

4.3. Automatic Evaluation of Translation Quality

In general, it is believed that translation between English and Japanese requires a large *distortion limit* (dl), which restricts how far a phrase can move. Some SMT researchers working on E-J or J-E translation use dl = –1 (unlimited) as a default value, and this takes a long translation time.

For PATMT J-E translation, Katz-Brown and Collins [2008] showed that dl = unlimited was the best in Japanese-to-English translation, but dl = unlimited took 37.2 seconds per sentence while dl = 6 took 5.0 seconds. That is, dl = unlimited was more than seven times slower than dl = 6.

For PATMT E-J translation, Moses* performed the best in terms of single reference BLEU, and this system also used dl = unlimited. However, Kumai et al. [2008] reported that they achieved the best result "when the distortion limit was 20 instead of –1." We tried to reproduce Moses* and ran MERT with `msd-bidirectional-fe` and dl = –1. This training took 35 hours. On the other hand, MERT of the proposed method with a distance-based model dl = 5 took only seven hours.

Table II compares the output of the proposed method and that of Moses* distributed by NTCIR7 PATMT organizers. Following the PATMT overview article [Fujii et al. 2008], we segmented the output Japanese sentences with ChaSen[3] and measured the quality with Bleu Kit[4]. However, how Japanese sentences are normalized is not explicitly described in the overview paper. Here, we converted hankaku (half-width) characters into zenkaku (full-width) characters.

We should pay attention to the difference of segmenters in this evaluation. Sophisticated MT systems tend to use in-house segmenters tuned for translation. Different segmenters give different segmentation results. One segmenter may treat "nihondenshindenwakabushikigaisha" (Nippon Telegraph and Telephone Corporation) as one word. Another segmenter may segment it into five words. For fair evaluation, we should use the same segmenter for both reference sentences and MT outputs.

Direct output of Moses follows the segmenter used in the training phase and we should not compare it directly with reference sentences segmented with ChaSen. We simply removed spaces between Japanese words in the Moses output to get final Japanese sentences, and then applied ChaSen to the sentences.

Table II shows that even dl = 0 (no reordering in Moses) gained better scores than Moses*. We compared proposed method (dl = 5) with Moses* by bootstrap resampling [Koehn 2004], and our proposed method won 1,000 times among 1,000 trials.

Without case marker seed words, the scores drop. However, this does not mean loss of case markers. Even without seed words, case markers are often inserted. By comparing the output translations, we found that boundaries of adjacent verb arguments becomes explicit by the seed words, and this fact leads to better translation.

[3] http://chasen-legacy.sourceforge.jp

[4] http://www.mibel.cs.tsukuba.ac.jp/norimatsu/bleu_kit/

Table II. Automatic Evaluation of Translation Quality

dl	BLEU	ROUGE-L	IMPACT	WER	PER	TER
	Larger is better			Smaller is better		
Moses*, the baseline method (NTCIR-7 PATMT Best BLEU score achiever)						
∞	**0.3063**	**0.4019**	**0.4022**	**0.7590**	**0.3925**	**0.5992**
Proposed Method						
0	0.3359	0.5049	0.4724	0.6365	0.3829	0.5261
1	0.3354	0.5051	0.4729	0.6362	0.3816	0.5255
2	0.3338	0.5030	0.4716	0.6363	0.3830	0.5266
3	**0.3361**	0.5062	0.4735	0.6354	**0.3813**	**0.5250**
4	0.3347	0.5045	0.4724	0.6361	0.3842	0.5271
5	0.3356	**0.5065**	**0.4736**	**0.6350**	0.3814	0.5251
6	0.3342	0.5032	0.4719	0.6383	0.3840	0.5291
7	0.3331	0.5044	0.4717	0.6382	0.3849	0.5287
9	0.3351	0.5049	0.4723	0.6367	0.3833	0.5269
12	0.3327	0.5040	0.4716	0.6383	0.3827	0.5282
Proposed Method Without Case Marker Seeds						
0	0.3150	0.4839	0.4579	0.6597	**0.3986**	0.5518
1	0.3118	0.4815	0.4555	0.6634	0.4033	0.5565
2	0.3119	0.4798	0.4551	0.6641	0.4036	0.5574
3	0.3136	0.4794	0.4557	0.6640	0.4056	0.5574
4	**0.3154**	0.4835	0.4575	0.6597	0.4016	0.5526
5	0.3141	0.4822	0.4571	0.6614	0.4026	0.5548
6	0.3159	**0.4855**	**0.4589**	**0.6579**	0.3996	**0.5506**
7	0.3110	0.4799	0.4557	0.6623	0.4040	0.5563
Proposed Method Without `repos`						
0	0.3161	0.4900	0.4617	0.6524	0.3886	0.5449
1	0.3173	0.4918	0.4626	0.6513	0.3867	0.5433
2	0.3160	0.4911	0.4610	0.6539	0.3905	0.5471
3	0.3190	0.4924	0.4625	0.6515	0.3865	0.5418
4	0.3199	0.4919	0.4628	0.6509	0.3887	0.5429
5	0.3198	**0.4939**	**0.4648**	0.6482	**0.3824**	**0.5367**
6	0.3160	0.4886	0.4604	0.6556	0.3890	0.5463
7	**0.3205**	0.4932	0.4647	**0.6472**	0.3843	0.5383
Proposed Method With Semantic Heads						
0	**0.3297**	**0.4972**	**0.4676**	**0.6431**	0.3877	**0.5334**
1	0.3262	0.4955	0.4657	0.6453	0.3892	0.5360
2	0.3258	0.4957	0.4653	0.6459	0.3906	0.5374
3	0.3280	0.4962	0.4664	0.6452	0.3885	0.5354
4	0.3277	0.4966	0.4666	0.6445	0.3885	0.5346
5	0.3260	0.4954	0.4656	0.6447	0.3873	0.5344
6	0.3279	0.4967	0.4671	0.6455	**0.3859**	0.5338
7	0.3264	0.4949	0.4667	0.6452	0.3876	0.5350

Without `repos`, the scores drop again. Using semantic heads is not as good as using syntactic heads.

For these ablation experiments, we applied two-tailed binomial test to automatic evaluation measures, WER, ROUGE-L, and IMPACT, which give sentence-level scores.

Table III. Segmenters in Evaluation

Translator	Segmenter	BLEU	ROUGE-L	IMPACT	WER	PER	TER
Proposed dl = 5	ChaSen	0.3356	0.5065	0.4736	0.6350	0.3814	0.5251
Proposed dl = 5	MeCab	0.3343	0.5071	0.4739	0.6325	0.3805	0.5237
Moses*	ChaSen	0.3063	0.4019	0.4022	0.7590	0.3925	0.5992
Moses*	MeCab	0.3054	0.4005	0.4046	0.7536	0.3837	0.5929

Table IV. Performance of Our Implementation of Related Systems
("MCS" Stands for Max-Chart-Span)

	BLEU	ROUGE-L	IMPACT	WER	PER	TER
tree-to-string mcs = 20	0.2421	0.3896	0.3926	0.7481	0.4327	0.6291
tree-to-string mcs = ∞	0.2450	0.3886	0.3892	0.7770	0.4646	0.6592
Xu-like dl = 3	0.2554	0.4052	0.4034	0.7438	0.4303	0.6222

As for dl = 0, the introduction of case marker seed words led to statistically significant improvements of these measures with $p < 10^{-12}$. The introduction of repos also yielded statistically significant improvements of these measures with $p < 10^{-6}$. The use of semantic heads degraded WER and IMPACT with $p < 0.03$ but the degradation of ROUGE-L was not statistically significant ($p = 0.075$).

On the other hand, position-independent word error rate (PER), which completely disregards word order, does not change very much. These facts indicate that our method improves word order, which is the most important problem in E-J translation.

In order to examine the effect of the difference of segmenters in the training phase and the evaluation phase, we also report the scores when we used MeCab as the evaluation segmenter. Table III shows the result. According to this table, translation quality scores did not change very much.

Our method is closely related to tree-to-string systems [Huang et al. 2006; Liu and Gildea 2008; Wu et al. 2010]. For comparison, we used Moses' tree-to-string model with Enju. Table IV summarizes their scores as well as our Xu-like system. We have not succeeded in showing their effectiveness in this NTCIR-7 PATMT EJ task yet.

5. DISCUSSION

Our method used an HPSG parser, which obtains rich information but is difficult to build. It is much easier to build word dependency parsers and Penn Treebank-style parsers. Here, we describe what happened when we used these parsers. When we used these parsers, we need additional rules like Xu et al. [2009].

5.1. Syntactic Head-Based Word Dependency Parsers

When we use an ordinary (syntactic head-based) word-dependency parser, we assume that the modified words are heads, and the Head Finalization rule is rephrased as "move modified words after modifiers."

As we described above, Xu et al. [2009] used semantic heads. Even when we use a syntactic head-based dependency parser, we encountered their 'excessive movement' problem. A straightforward application of their rules changes

(0) John hit the ball but Sam threw the ball.

into

(1) John the ball but Sam the ball threw hit.

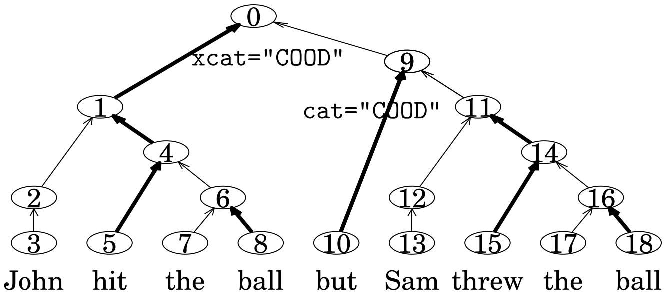


Fig. 5. Head Finalization does not mix up clauses.

Here, the two clauses are mixed up. To prevent this, they disallowed any movement across punctuation and conjunctions. We mentioned this in Section 4.2. Then they achieved a better result:

```
(2) John the ball hit but Sam the ball threw.
```

When we used Enju, these clauses were not mixed up. Enju-based Head Finalization achieved the same word order as Example (2):

```
(3) John _va1 ball _va2 hit but Sam _va0 ball _va2 throw.
```

Figure 5 shows Enju's parse tree. Because of our coordination exception rule, "Sam" becomes the main subject. When Head Finalization swaps the children of a mother node, the children do not move beyond the range of the mother node. Therefore, Head Finalization based on Enju does not mix up the first clause "John hit the ball" covered by Node 1 with the second clause "Sam threw the ball" covered by Node 11. Thus, non-terminal nodes in Enju's output are useful to protect clauses.

On the other hand, according to our syntactic head-based word dependency parser, "threw" is a modifier of "hit." Therefore, "hit" is moved after "threw" just like Example (1), and the two clauses become mixed up. Consequently, we need a rule like Xu et al.'s [2009] to avoid this excessive movement problem.

5.2. Penn Treebank-Style Parsers

We also tried Charniak-Johnson's parser [Charniak and Johnson 2005]. PyInputTree[5] gives heads. Enju outputs at most two children per mother node, but Penn Treebank-style parsers do not have such a limitation on the number of children, which causes a problem.

When we use Enju, "This toy is popular in Japan" is reordered as "This toy _va0 Japan in popular is." Its monotonic translation is natural: "kono omocha wa nihon de ninki ga aru."

On the other hand, Charniak-Johnson's parser outputs the following S-expression for this sentence (we added asterisks (*) to indicate heads).

```
(S (NP (DT This) (NN* toy))
   (VP* (AUX* is) (ADJP (JJ* popular)) (PP (IN* in) (NP (NNP* Japan)))))
```

This VP has three children: AUX, ADJP, and PP. Simply moving heads to the end breaks the adjacency of "is" and "popular": "this toy _va0 popular Japan in is." If we translate this monotonically, "Japan" becomes "popular" instead of "toy". We can solve this problem by using Xu et al.'s children reversal. This reconnects is and popular

[5] http://www.cs.brown.edu/˜dmcc/software/PyInputTree/

and we obtain "This toy _va0 Japan in popular is" from the following reversed S-expression.

```
(S (NP (DT This) (NN* toy))
   (VP* (PP (IN* in) (NP (NNP* Japan))) (ADJP (JJ* popular)) (AUX* is)))
```

5.3. Limitation and Advantage of Head Finalization

Head Finalization achieves a good first approximation of Japanese word order in spite of its simplicity. Since it is just a rule-based method and does not use any machine learning methods, its reordering performance does not improve even when the amount of training data increases. However, Moses' distance-based reordering helps its reordering performance.

Statistical machine translation depends on the assumption that we can obtain a large parallel corpus. This assumption holds for major European languages, but it is often difficult to satisfy this assumption for Asian languages. We hope Head Finalization helps building machine translation systems for such Asian languages that follow head-final word order.

One weak point of current Head Finalization is negatively translated noun modifiers such as "no" or "few" in long sentences. When an HFE sentence has such a word, it is difficult to translate the sentence monotonically. For instance, "I have no time" becomes "I _va0 no time _va2 have." Its monotonic translation is "∗watashi wa nai jikan wo motteiru," but this sentence is not acceptable. An acceptable literal translation is "(watashi niwa) jikan ga nai." Here, the English word "no" corresponds to "nai" at the end of the Japanese translation. When the sentence is long, we will need a large distortion limit. One solution is to rephrase the English sentence with "not" that modifies a verb or an auxiliary verb.

On the other hand, the advantage of Head Finalization other than simplicity is its granulaity. Some alternative preprocessing methods simply divide long sentences into short clauses, and translate each clause. These methods are also effective because short sentences do not require a large distortion limit. Kumai et al. [2008] divided sentences at commas. Sudoh et al. [2010] used Enju to parse English sentences, and removed subordinate clauses marked with "S" (footnote 1 in their p. 418) and substituted them for place-holders. Then, each clause is translated independently, and reconstructed by back-substituting the place-holders for corresponding translated clauses.

However, these division methods do not work for the following article title [Nakachi et al. 2010] at all because it has neither a comma nor "S."

```
A phase II study of induction chemotherapy with gemcitabine plus S-1
followed by chemoradiotherapy for locally advanced pancreatic cancer
```

Head Finalization works even in this case, and obtains the following HFE.

```
locally advanced pancreatic cancer for chemoradiotherapy by followed
gemcitabine plus S-1 with induction chemotherapy of phase II study
```

It can be monotonically translated into Japanese.

kyokusho shinko suigan ni-taisuru houshasen-kagaku-ryouhou wo heiyou-shita
gemushitabin purasu S-1 deno dounyuu kagakuryouhou no dainisou shiken

Wu et al. [2010] also used Enju for English-to-Japanese translation. Their method extracts about 100 million rules on the basis of Galley et al. [2004]. Perhaps, the number of rules is unnecessarily large because of parser errors and nonstandard translations in the training data.

Genzel [2010] proposed a machine learning-based approach that kept the number of rules small and obtained only 40-50 rules. For Japanese, Genzel showed that his method obtained a slightly better BLEU score (0.2912) than Xu et al.'s [2009] manual rule set (0.2878).

Our article clarified that even one reordering rule gives a good approximation of the English-to-Japanese reordering and even this simple rule really improves the translation quality in terms of various automatic evaluation scores.

6. CONCLUSION

To solve the word-order problem in English-to-Japanese translation, we introduced a new reordering rule called Head Finalization. We can implement word reordering in an abstract level, and the reordering rule is very simple compared to those in related work. Automatic evaluation scores showed that our method significantly improved the translation quality.

However, Head Finalization requires a sophisticated HPSG parser such as Enju, and Enju's source code is not publicly available yet. We discussed the problems of other parsers and clarified that the use of syntactic heads and the binary tree constraint is essential.

Our future work is to build our own efficient and accurate dependency parser that outputs what we need in our method and to apply this method to other head-final languages such as Korean and Turkish.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Yusuke Miyao for his useful advice on the usage of Enju. We thank the audience and the reviewers for their comments.

REFERENCES

CHARNIAK, E. AND JOHNSON, M. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'05)*. 173–180.

COLLINS, M., KOEHN, P., AND KUCEROVA, I. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'05)*.

DE MARNEFFE, M.-C., MACCARTNEY, B., AND MANNING, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Language Resources and Evaluation Conference (LREC'06)*. 449–454.

ECHIZEN-YA, H., EHARA, T., SHIMOHATA, S., FUJII, A., UTIYAMA, M., YAMAMOTO, M., UTSURO, T., AND KANDO, N. 2009. Meta-evaluation of automatic evaluation methods for machine translation using patent translation data in NTCIR-7. In *Proceedings of the 3rd Workshop on Patent Translation (WPT'09)*. 9–16.

FUJII, A., UTIYAMA, M., YAMAMOTO, M., AND UTSURO, T. 2008. Overview of the patent translation task at the NTCIR-7 workshop. In *Working Notes of the NII Test Collection for Information Retrieval Workshop Meeting (NTCIR'08)*. 389–400.

GALLEY, M., HOPKINS, M., KNIGHT, K., AND MARCU, D. 2004. What's in a translation rule? In *Proceedings of the North American Chapter of the Association of Computational Linguistics (NAACL'04)*. 273–280.

GENZEL, D. 2010. Automatically learning source-side reordering rules for large-scale machine translation. In *Proceedings of the International Conference on Computational Linguistics (COLING'10)*. 376–384.

HONG, G., LEE, S.-W., AND RIM, H.-C. 2009. Bridging morpho-syntactic gap between source and target sentences for English-Korean statistical machine translation. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP'09)*. 233–236.

HUANG, L., KNIGHT, K., AND JOSHI, A. 2006. A syntax-directed translator with extended domain of locality. In *Proceedings of Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing (CHPJISLP'06)*. 1–8.

ISOZAKI, H., SUDOH, K., TSUKADA, H., AND DUH, K. 2010. Head finalization: A simple reordering rule for SOV languages. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR (SMT'10)*. 250–257.

IZUHA, T., KUMANO, A., AND KURODA, Y. 2008. Toshiba rule-based machine translation system at NTCIR-7 PAT MT. In *Working Notes of the NII Test Collection for Information Retrieval Workshop Meeting (NTCIR'08)*. 430–434.

KATZ-BROWN, J. AND COLLINS, M. 2008. Syntactic reordering in preprocessing for Japanese → English translation: MIT system description for NTCIR-7 patent translation task. In *Working Notes of the NII Test Collection for Information Retrieval Workshop Meeting (NTCIR'08)*.

KENDALL, M. G. 1975. *Rank Correlation Methods*. Charles Griffin.

KOEHN, P. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*. 388–395.

KOEHN, P. 2009. *Statistical Machine Translation*. Cambridge University Press.

KOEHN, P. 2010. *MOSES, Statistical Machine Translation System, User Manual and Code Guide*. www.statmt.org/moses/manual/manual.pdf.

KUMAI, H., SEGAWA, H., AND MORIMOTO, Y. 2008. NTCIR-7 patent translation experiments at Hitachi. In *Working Notes of the NII Test Collection for Information Retrieval Workshop Meeting (NTCIR'08)*. 441–444.

LEE, Y.-S., ZHAO, B., AND LUO, X. 2010. Constituent reordering and syntax models for English-to-Japanese statistical machine translation. In *Proceedings of the International Conference on Computational Linguistics (COLING'10)*. 626–634.

LI, C.-H., ZHANG, D., LI, M., ZHOU, M., LI, M., AND GUAN, Y. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'07)*. 720–727.

LIN, C.-Y. AND OCH, F. J. 2004. Automatic evaluation of translation quality using longest common subsequences and skip-bigram statistics. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'04)*. 605–612.

LIU, D. AND GILDEA, D. 2008. Improved tree-to-string transducer for machine translation. In *Proceedings of the Workshop on Statistical Machine Translation (SMT'08)*. 62–69.

MIYAO, Y. AND JUN'ICHI TSUJII. 2008. Feature forest models for probabilistic HPSG parsing. *Comput. Linguist. 34*, 1, 35–80.

MIYAO, Y. AND TSUJII, J. 2005. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'05)*. 83–90.

NAKACHI, K., FURUSE, J., KINOSHITA, T., KAWASHIMA, M., AND AND, H. I. 2010. A phase II study of induction chemotherapy with gemcitabine plus S-1 followed by chemoradiotherapy for locally advanced pancreatic cancer. *Cancer Chemo. Pharmacol. 66*, 3, 527–534.

NAKAZAWA, T. AND KUROHASHI, S. 2008. Kyoto-u: Syntactical EBMT system for NTCIR-7 patent translation task. In *Working Notes of the NTCIR Workshop Meeting (NTCIR'08)*. 401–408.

NGUYEN, T. P. AND SHIMAZU, A. 2006. Improving phrase-based statistical machine translation with morphosyntactic transformation. *Mach. Trans. 20*, 3, 147–166.

OCH, F. J. AND NEY, H. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist. 29*, 1, 19–51.

PAPINENI, K., ROUKOS, S., WARD, T., AND ZHU, W.-J. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'02)*. 311–318.

POLLARD, C. AND SAG, I. A. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

QUIRK, C., MENEZES, A., AND CHERRY, C. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'05)*. 271–279.

SNOVER, M., DORR, B., SCHWARTZ, R., MICCIULLA, L., AND MAKHOUL, J. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas (AMTA'06)*.

SU, K.-Y., WU, M.-W., AND CHANG, J.-S. 1992. A new quantitative quality measure for machine translation systems. In *Proceedings of the International Conference on Computational Linguistics (COLING'92)*. 433–439.

SUDOH, K., DUH, K., TSUKADA, H., HIRAO, T., AND NAGATA, M. 2010. Divide and translate: Improving long distance reordering in statistical machine translation. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR (SMT'10)*. 418–427.

TOUTANOVA, K. AND SUZUKI, H. 2007. Generating case markers in machine translation. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT-NAACL'07)*. 49–56.

WATANABE, T., TSUKADA, H., AND ISOZAKI, H. 2008. NTT SMT System 2008 at NTCIR-7. In *Working Notes of the NTCIR Workshop Meeting (NTCIR'08)*. 420–422.

WU, D. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist. 23*, 3, 377–403.

WU, X., MATSUZAKI, T., AND TSUJII, J. 2010. Fine-grained tree-to-string translation rule extraction. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'10)*. 325–334.

XIA, F. AND MCCORD, M. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the International Conference on Computational Linguistics (COLING'04)*. 508–514.

XU, P., KANG, J., RINGGAARD, M., AND OCH, F. 2009. Using a dependency parser to improve SMT for Subject-Object-Verb languages. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT-NAACL'09)*. 245–253.

YAMADA, K. AND KNIGHT, K. 2001. A syntax-based statistical translation model. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL'01)*. 523–530.

Received March 2011; revised June 2011; accepted August 2011