Extracting Pre-ordering Rules from Chunk-based Dependency Trees for Japanese-to-English Translation

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Masaaki Nagata
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai Seika-cho, Soraku-gun Kyoto 619-0237 Japan
{wu.xianchao,sudoh.katsuhito,kevin.duh,tsukada.hajime,nagata.masaaki}@lab.ntt.co.jp

Abstract

Word ordering remains as an essential problem for translating between languages with substantial structural differences, such as SOV and SVO languages. In this paper, we propose to use source chunk-based dependency trees to extract pre-ordering rules, which are customized for translating from Japanese (SOV) to English (SVO). In order to obtain a fine-grained classification of the reordering phenomena, Japanese function words and punctuation marks are included in the pre-ordering rules as additional lexical clues. We further prune the pre-ordering rule set by referring the predicate-argument structures of the target language side. Experimental results are reported for large-scale Japanese-to-English translation, showing a significant improvement of 1.48 BLEU points compared with the baseline SMT system.

1 Introduction

Statistical machine translation (SMT) suffers from an essential word-ordering problem for translating between languages with substantial structural differences, such as between Japanese which is a typical subject-object-verb (SOV) language and English which is a subject-verb-object (SVO) language.

Numerous approaches have been consequently proposed to tackle this word-order problem, such as lexicalized reordering methods (Tillman, 2004; Kumar and Byrne, 2005), syntax-based models (Galley et al., 2004; Chiang, 2005; Liu et al., 2006; Mi et al., 2008), and pre-ordering ways.

This paper tackles the word-order problem in a pre-ordering way. Through the usage of a sequence of pre-ordering rules, the word order of an original source sentence is (approximately) changed into the word order of the target sentence. Here, the pre-ordering rules can be manually or automatically extracted. For manual extraction of pre-ordering rules, linguistic background and expertise are required for pre-determined language pairs, such as for German-English (Collins et al., 2005), Chinese-to-English (Wang et al., 2007), Japanese-to-English (Katz-Brown and Collins, 2007), and English-to-SOV languages (Xu et al., 2009).

The goal in this paper, however, is to learn pre-ordering rules from parallel data in an *automatic* way. Under this motivation, pre-ordering rules can be extracted in a language-independent manner. A number of researches follow this automatic way. For example, in (Xia and McCord, 2004), a variety of heuristic rules were applied to bilingual parse trees to extract pre-ordering rules for French-English translation. Rottmann and Vogen (2007) learned reordering rules based on sequences of part-of-speech (POS) tags, instead of parse trees. A chunk-level pre-ordering approach was described by Zhang et al. (2007). The reordering rules were first automatically learned from source-side chunks and word alignments and then used to generate a reordering lattice for each sentence. Constituency trees based on context free grammar and context sensitive grammar were used by Lee et al. (2010) to extract pre-ordering rules for English-to-Japanese translation. Syntactic trees were used by Visweswariah et al. (2010) to extract pre-ordering rules for English-to-

Hindi, Spanish, and French translation. Dependency trees were used by Genzel (2010) to extract source-side reordering rules for translating languages from SVO to SOV, etc..

Different from these former approaches, we make use of *chunk-based dependency trees* for extracting pre-ordering rules. That is, the semantic dependencies are constructed among chunks instead of single words. By taking chunks as the basic unit for pre-ordering, we hope to capture the reordering patterns among chunks yet keep the relations of the words inside chunks. Our proposal includes the following novel ideas:

- making use of source chunk-based dependency trees so that pre-ordering rules can be extracted in a *abstract level*;
- including Japanese function words[1] and punctuation marks as additional *lexical clues* in the pre-ordering rules to yield a *fine-grained classification* of the reordering phenomena; and,
- pruning the pre-ordering rule set by taking predicate-argument structures (PASs) of the target language side as additional constraints.

By taking target PASs as additional constraints, similar idea was proposed in (Gao and Vogel, 2011). Target-side semantic role labels (SRLs) were used by them to extract SRL-aware synchronous context-free grammar (SCFG) rules for assisting hierarchical phrase-based translation. However, the difference from our work is obvious, we extract monolingual pre-ordering rules instead of bilingual SCFG rules to be dynamically used in CKY decoding.

In this paper, we use the Cabocha v0.53[2] (Kudo and Matsumoto, 2002) dependency parser to generate the chunk-level dependency trees for the Japanese sentences. Japanese function words and auxiliary verbs are automatically identified and included in the output of Cabocha. Note that similar parser has been used by Katz-Brown and Collins (2007). However, instead of dynamically referring the word alignments for pre-ordering rule extraction and application as done in this paper, they simply manually defined several pre-ordering rules which tent out to be too subjective to improve the final translation accuracy.

Following (Wu et al., 2010; Isozaki et al., 2010b), we use the head-driven phrase structure (HPSG) parser Enju to generate the PASs of English sentences. HPSG (Pollard and Sag, 1994; Sag et al., 2003) is a lexicalist grammar framework. In HPSG, linguistic entities such as words and phrases are represented by a data structure called a *sign*. A sign gives a factored representation of the syntactic features of a word/phrase, as well as a representation of their *semantic content* which corresponds to PASs.

In order to extract and apply pre-ordering rules in a ordered way, we first construct a constituency tree based on a given dependency tree. Then, we describe a pre-ordering rule extraction algorithm through a bottom-up traversal of the constituency tree. Finally, we apply the pre-ordering rules into a constituency tree to yield a target-word-order alike source sentence.

The remaining of this paper is organized as follows. In the next section, we describe the algorithms for extracting and applying pre-ordering rules. For intuitive understanding, we use a real example for explanation. In Section 3, we design experiments on large-scale Japanese-to-English translation to testify our proposal. Employing Moses (Koehn et al., 2007), we show that our proposal can significantly improve BLEU score of 1.48 points compared with using the original Japanese sentences. We finally conclude this paper by summarizing our proposal and the experiment results in Section 4.

2 Pre-ordering Rule Extraction and Application

2.1 An example

For an intuitive understanding of our proposed approaches, Figure 1 shows a word-aligned[3] dependency-HPSG tree pair for Japanese-to-English translation. In the Japanese side (bottom of the figure), arrows in thin lines represent the dependencies between chunks and their semantic heads. Inside the chunks, there are frequently a head phrase followed

[1] Please refer to (Wu et al., 2011) and (Martin, 1975) for the detailed definition and categories of Japanese function words and auxiliary verbs.

[2] http://chasen.org/∼taku/software/cabocha/

[3] These word alignments are gained by running GIZA++ (Och and Ney, 2003) on the original parallel sentences.

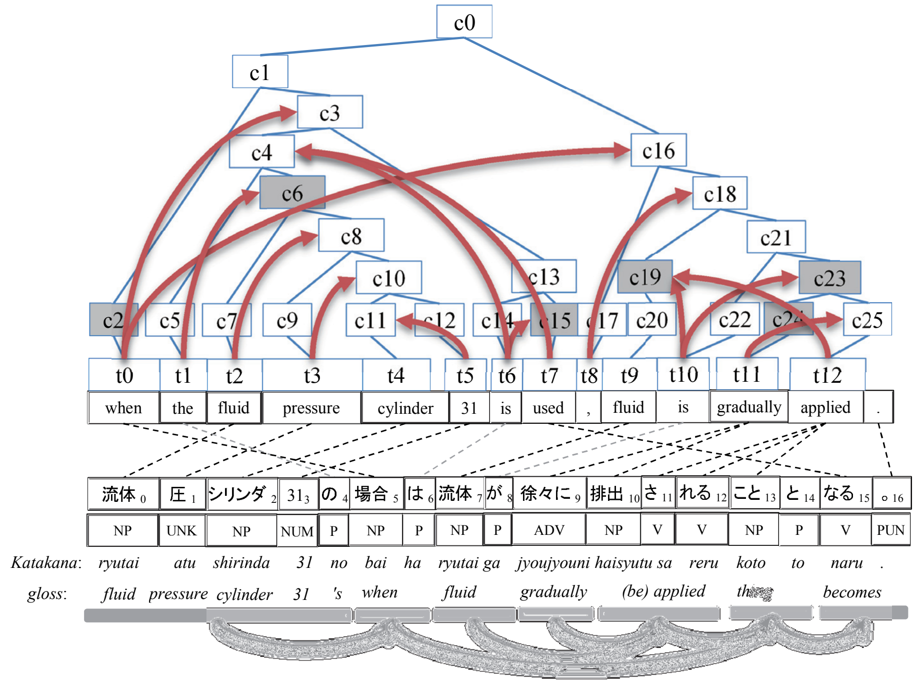

Figure 1: Illustration of a word-aligned dependency-HPSG tree pair for Japanese-English translation.

by some function words or punctuation marks. For example, the first chunk in the figure contains a noun phrase "ryutai atu shirinda 31" and a function word "no".

In the English side (top of the figure), predicate-argument structures among lexical nodes and their argument nodes in this HPSG tree are described by arrows in thick lines. For simplicity, we only draw the identifiers for the signs of the nodes in the HPSG tree. Note that the identifiers that start with 'c' denote non-terminal nodes (e.g., c0, c1), and the identifiers that start with 't' denote terminal nodes (e.g., t0, t1). In a complete HPSG forest given in (Wu et al., 2010), factored syntactic features are included in these terminal and non-terminal nodes.

2.2 Rule extraction algorithm

We mainly focus on the example shown in Figure 1 to express our algorithms. We first describe an algorithm to transfer a dependency tree into a constituency tree. The consideration behind is to extract and apply pre-ordering rules in a ordered way, such as a bottom-up traversal. In addition, we leave out Japanese function words and punctuation marks as special *lexical clues* to sub-categorize pre-ordering rules. Then, we briefly describe a pre-ordering rule extraction algorithm through a bottom-up traversal of the constituency tree. We sort the leaves of the tree fragments based on several heuristics. Finally, we describe the process of applying the pre-ordering rules into some new constituency trees to yield a target-word-order alike source sentence.

Figure 2 shows the way to change a dependency tree into a constituency tree. The algorithm is quite simple. Through a topological scan of the chunks, we create a POS node for each chunk. By creating these POS nodes, we are trying to extract an *abstract level* pre-ordering rule set. The POS label of the node takes the POS of the head (or dominant) phrase in the chunk. For example, for the first chunk in the figure, the POS sequence of the head phrase is "NP UNK NP NUM" where UNK means unknown word, thus we take "NP" as the label of the newly created POS node. We define verb has a higher priority than noun. For example, the POS sequence for the fifth trunk is "NP V V", thus we take "V" as the label for the POS node.

Also, during the scanning, every time we meet a head chunk (i.e., there are at least one in-arrow), we

1. create a non-terminal node with a label X;

2. connect the head chunk and all of its direct

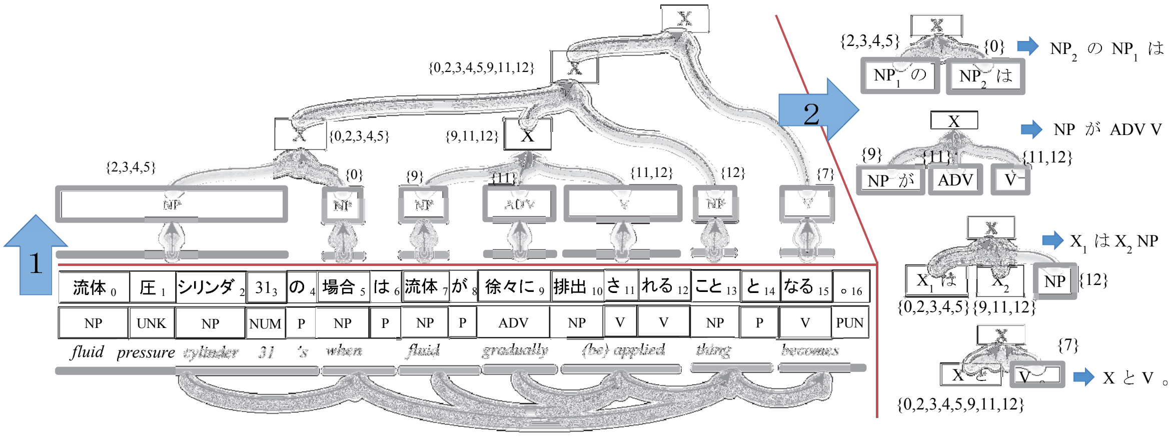

Figure 2: Illustration of 1) changing a dependency tree to a constituency tree, and 2) extracting of pre-ordering rules. The target spans of the nodes are drawn as well.

child chunks to this newly created non-terminal node X; and,

3. connect this X to the parent chunk (if exists) of current head chunk.

After building a constituency tree from the original chunk-based dependency tree, we can extract pre-ordering rules by a bottom-up traversal of the constituency tree. In this paper, we only take the POS nodes and the non-terminal X nodes into consideration. However, as will be shown in our experiments (refer to Table 1), there are more than 25% of particles in the Japanese sentences. We keep to use these particles as lexical clues in our pre-ordering rules. This means we only consider the relative positions among the head phrases in the chunks.

The pre-ordering rules (three monotonic rule and one reordering rule) extracted from the example in Figure 1 are listed in the right-hand-side of Figure 2. For similarity, we limit to extract pre-ordering rules that contain no more than two layers. This means we only include a non-terminal node and its direct child nodes in a pre-ordering rule. Through this limitation, we can constrain the pre-ordering rules in a minimal size. So that, during applying of the pre-ordering rules, we can retrieve available rules in a relatively fast way without generating numerous subtrees of current non-terminal node.

Based on pre-given words alignments, we compute the target spans of the nodes of the constituency tree, as shown in Figure 2. Considering there are overlapping among the target spans, we heuristically sort the target spans of the leaf nodes in a pre-ordering rule. Suppose there are two spans, named span A and span B:

- if more than half of numbers in A is bigger than the maximum number in B, or if more than half of numbers in B is smaller than the minimum number in A, then B $<$ A;

- if more than half of numbers in B is bigger than the maximum number in A, or if more than half of numbers in A is smaller than the minimum number in B, then A $<$ B.

In case of a tie (e.g., A={3,4,7,8}, B={5,6}), we keep the original order of A and B in the source-side sentence without any reordering.

As former mentioned, we prune pre-ordering rules by referring to the PASs of the target sentences. The constraint we use is to check whether the source words covered by the leaf nodes are aligned to a partial of some argument phrase(s) in the target PASs. That is, we keep a pre-ordering rule only if the leaf nodes of the left-hand-side tree are all aligned to some *complete* linguistic phrase(s) in the target language side. For example, for the pre-reordering rule in the top-right corner of Figure 2, its leaves are aligned to node c6 and c2, respectively. Since c6 here is an argument node and c2 is a predicate node, we take this rule to be legal.

2.3 Rule application algorithm

Dealing with the training data, we perform the reordering for each source Japanese sentence during the process of extracting pre-ordering rules. This is because the alignments and the target English sentences' predicate-argument structures are known beforehand. This strategy enables us to gain a local-optimal reordered Japanese sentence.

Another strategy is to first collect all the reordering rules from the training data, and then apply them to each parallel sentence again. This is a kind of "global" optimization strategy, since different parallel sentences are *sharing* their pre-ordering rules. However, this strategy may introduce too generalized pre-ordering rules to be applied to the specific individuals. Another problem of this strategy is that we will have to manage a n-best list to store the numerous possibilities for pre-reordering one Japanese sentence. In order to re-train the alignment, we have to pick only one reordered sentence from the n-best list. Based on these considerations and for simplicity, we perform local-optimal strategy in this paper.

In the development/test set, word alignments and PASs of target sentences are unknown. The rule application algorithm incudes the following steps:

1. change the dependency tree into a constituency tree;
2. apply the pre-ordering rules though a bottom-up traversal of the constituency tree, and keep a k-best list for each non-terminal node;
3. pick one reordered sentence from the k-best list of the root node of the constituency tree.

In order to pick the optimal reordered sentence, we train a n-gram language model using the reordered Japanese sentences in the training data and then select one sentence with the highest language model score[4]. We hope to use this selection strategy to balance the fluency of the words appearing both inside (i.e., in the head phrases) and outside (i.e., Japanese particles and punctuation) the chunks.

[4]Currently, we only used LM score for ranking the candidates. More features, such as the frequency of a rule, are suspected to be used for ranking in the future.

3 Experiments

3.1 Setup

We test our proposal by translating from Japanese to English. We use the NTCIR-9 English-Japanese patent corpus[5] (Utiyama and Isahara, 2007) as our experiment set. Since the reference set of the official test set has not been released yet, we instead split the original development set averagely into two parts, named dev.a and dev.b. In our experiments, we first take dev.a as our development set for minimum-error rate training (Och, 2003) and then report the final translation accuracies on dev.b. For direct comparison with other systems in the future, we use the configuration of the official baseline system[6]:

- Moses[7] (Koehn et al., 2007): revision = "3717" as the baseline decoder;
- GIZA++: giza-pp-v1.0.3[8] (Och and Ney, 2003) for first training word alignment using the original Japanese sentences for pre-ordering rule extraction, and then for retraining word alignments using the pre-ordered Japanese sentences;
- SRI LM[9] (Stolcke, 2002): version 1.5.12 for training a 5-gram language model using the target sentences in the total training set;
- Additional scripts[10]: for preprocessing English sentences and cleaning up too long (# of words > 40) parallel sentences;
- Japanese word segmentation: Mecab v0.98[11] with the dictionary of mecab-ipadic-2.7.0-20070801.tar.gz[12].

The statistics of the filtered training set, dev.a, and dev.b are shown in Table 1. The success parsing rate ranges from 98.7% to 99.3% by using Enju2.3.1. The averaged parsing time for each English sentence ranges from 0.30 to 0.48 seconds.

[5]http://ntcir.nii.ac.jp/PatentMT/
[6]http://ntcir.nii.ac.jp/PatentMT/baselineSystems
[7]http://www.statmt.org/moses/
[8]http://giza-pp.googlecode.com/files/giza-pp-v1.0.3.tar.gz
[9]http://www.speech.sri.com/projects/srilm/
[10]http://homepages.inf.ed.ac.uk/jschroe1/how-to/scripts.tgz
[11]http://sourceforge.net/projects/mecab/files/
[12]http://sourceforge.net/projects/mecab/files/mecab-ipadic/

	Train	Dev.a	Dev.b
# parallel sentences	2,032,679	1,000	1,000
# En words	48,322,058	31,890	31,935
Enju success parse rate	99.3%	98.9%	98.7%
parse time (sec./sent.)	0.30	0.38	0.48
# Jp words	53,865,629	37,066	35,921
# Jp function words	13,771,582	9,357	9,256
% Jp function words	25.6%	25.2%	25.8%

Table 1: Statistics of the parallel sentence sets used in the experiments.

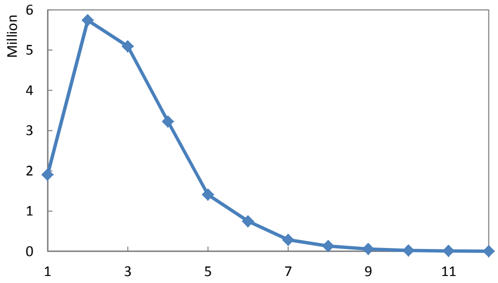

Figure 3: Distribution of the number of words in the chunks.

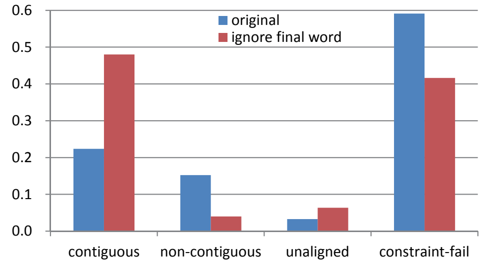

Figure 4: Comparison of the alignments of chunks.

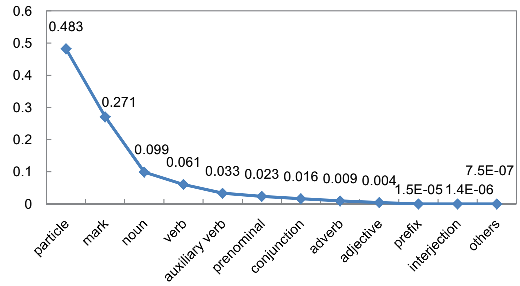

Figure 5: Distribution of the percentage of the final words' POSs in the chunks.

3.2 Statistics of the chunks

As shown in Figure 3, we investigated the number of words included in the Japanese chunks. There are 93.0% chunks that contain no more than five words. We thus train a 5-gram LM on the reordered Japanese sentences for reranking the k(=100)-best pre-ordering candidate sentences in the dev./test sets.

We took chunks as the basic unit for reordering. A consequent question is that, how well are the Japanese chunks aligned to the target sentences?

To answer this question, we further investigated the statistics of the alignments of the chunks. We classify the alignments into four types: contiguous (i.e., a source chunk is aligned to a contiguous target phrase and the target phrase is not aligned to any source words other than this source chunk), non-contiguous, un-aligned, and align-constraint-fail (i.e., a source chunk is aligned to some target words yet these target words are also aligned to some source words that are not included in this source chunk). It is easy to understand that we prefer more contiguous alignments and less align-constraint-fail alignments.

Figure 4 shows the statistics of the alignments of the source chunks. Unfortunately, there are only 22.4% chunks whose alignments are contiguous and 59.1% chunks whose alignments broke the alignment constraints. Is there a way to refine the original alignments?

To answer this question, we investigated the distribution of the POS of the word that appears at the right-most side (denote as "final words", hereafter) of the chunks, as shown in Figure 5. From this figure, we can see that *particle* and *mark* dominantly occur 48.3% and 27.1% of the total POS types. Thus, it is reasonable for us to take them as a special *lexical clue* to be used apparently in the pre-ordering rules. Indeed, when we ignore the alignments of the final words in the chunks[13], the statistics of the chunks' alignments tent to be significantly better, as shown in Figure 4. Similar idea of ignoring the alignments of the Japanese function words has been applied by us in forest-based translation (Wu et al., 2011).

Now, by removing the alignments of the final words in the chunks, there are 48.0% chunks that aligned to contiguous target phrases. The percent-

[13]We perform this operation only if there are no less than two words in the chunk. Also, note that even we delete the alignments for all the final words, we only take particles and punctuation marks as the lexical clues in our pre-ordering rules.

Rule type	# total	# reorder	%
no fw	61,721	57,975	93.9%
fw	797,059	590,024	74.0%
fw-pas	682,837	534,378	78.3%

Table 2: Statistics of the pre-ordering rules.

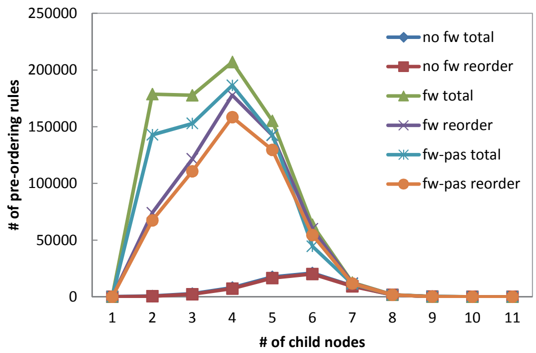

Figure 6: Distribution of the pre-ordering rules.

age of chunks that broke the alignment constraints dropped from 59.1% to 41.7%. Another interesting benefit is that, the percentage of chunks that aligned to non-contiguous target phrases also dropped from 15.2% to 4.0%. Since we ignored the alignments of the final words, the percentage of unaligned chunks increased from 3.3% to 6.4%.

3.3 Statistics of the pre-ordering rules

The statistics of our pre-ordering rules are shown in Table 2. Here, "no fw" represents the rule set that was extracted 1) without deleting the alignments of the final words in the Japanese chunks, and 2) without taking Japanese particles/marks as additional lexical clues. Easy to see that we extract more generalized rules with a small size. In contrast, "fw" represents the rule set that was extracted using our proposal of taking the functional words as lexical clues. Through this way, we obtained a fine-grained pre-ordering rule set. Third, "fw-pas" denotes the rule set that was extracted using both final words and PAS constraints. Under this constraint, the number of rules was pruned from 797,059 to 682,837. From the table, we can see that there are no less than 74% reordering rules. These reflect that reordering is essential for translation Japanese (SOV) into English (SVO). For intuitive comparison, the distributions of the "size" (i.e., the number of child nodes in the left-hand-side tree fragment) of the pre-ordering rules are shown in Figure 6.

Source sent.	BLEU	RIBES
original sentences	0.2602	0.6614
no fw reordered	0.2567	0.6670
fw reordered	0.2668*	0.6812
fw-pas reordered	**0.2750****	**0.6908**

Table 3: Translation accuracies by using the original Japanese sentences or the pre-ordered Japanese sentences. Here, * = $p < 0.05$ and ** = $p < 0.01$.

3.4 Results

Table 3 shows the final translation accuracies under BLEU score (Papineni et al., 2002) and RIBES[14], i.e., the software implementation of Normalized Kendall's τ as proposed by (Isozaki et al., 2010a) to automatically evaluate the translation between distant language pairs based on rank correlation coefficients and significantly penalizes word order mistakes. First, using "no fw" rule set for pre-ordering did not benefit BLEU score, even RIBES was increased slightly. Though the comparison with using "fw" rule set, we can see that to pre-order among head phrases in the chunks instead of the final words does significantly ($p < 0.05$) improve the translation accuracy. Finally, using "fw-pas" rule set, we significantly ($p < 0.01$) improved 1.48 BLEU points. The effectiveness of our proposal is also testified by comparing the improvement of the scores of RIBES.

4 Conclusion

We have proposed a pre-ordering approach by making use of chunk-based dependency trees. The pre-ordering rules record the relative source-target position mapping among the head phrases in the Japanese chunks. We proposed to include Japanese function words and punctuation marks in the pre-ordering rules as lexical clues and prune pre-ordering rules by linguistic constraints from target PASs. Employing Moses (Koehn et al., 2007), our proposal significantly improved BLEU score of 1.48 points compared with using the original Japanese sentences. We finally argue that our proposed approach is not difficult to be re-implemented and all the resources used in this paper can be freely downloaded or obtained.

[14]Code available at http://www.kecl.ntt.co.jp/icl/lirg/ribes

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc.of ACL*, pages 263–270.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc.of ACL*, pages 531–540.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proc.of HLT-NAACL*.

Qin Gao and Stephan Vogel. 2011. Utilizing target-side semantic role labels to assist hierarchical phrase-based machine translation. In *Proc. of 5th workshop on SSST*, pages 107–115.

Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In *Proc.of COLING*, pages 376–384.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010a. Automatic evaluation of translation quality for distant language pairs. In *Proc.of EMNLP*, pages 944–952.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010b. Head finalization: A simple reordering rule for sov languages. In *Proc.of WMT-MetricsMATR*, pages 244–251.

Jason Katz-Brown and Michael Collins. 2007. Syntactic reordering in preprocessing for japanese-english translation: Mit system description for ntcir-7 patent translation task. In *Proc.of NTCIR-7 Workshop Meeting*, pages 409–414.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *the ACL 2007 Demo-Poster*, pages 177–180.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proc.of CoNLL-2002*, pages 63–69. Taipei, Taiwan.

Shankar Kumar and William Byrne. 2005. Local phrase reordering models for statistical machine translation. In *Proc.of HLT-EMNLP*, pages 161–168.

Young-Suk Lee, Bing Zhao, and Xiaoqian Luo. 2010. Constituent reordering and syntax models for english-to-japanese statistical machine translation. In *Proc. of COLING*, pages 626–634.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment templates for statistical machine transaltion. In *Proc.of COLING-ACL*, pages 609–616, Sydney, Australia.

Samuel E. Martin. 1975. *A Reference Grammar of Japanese*. New Haven, Conn.: Yale University Press.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proc.of ACL-08:HLT*, pages 192–199, Columbus, Ohio.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc.of ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc.of ACL*, pages 311–318.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Kay Rottmann and Stephan Vogel. 2007. Word reordering in statistical machine translation with a pos-based distortion model. In *Proc.of TMI*, pages 171–180, Skovde, Sweden.

Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. Number 152 in CSLI Lecture Notes. CSLI Publications.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proc.of ICSLP*, pages 901–904.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proc. of HLT-NAACL*, pages 101–104.

Masao Utiyama and Hitoshi Isahara. 2007. A japanese-english patent parallel corpus. In *Proc.of MT Summit XI*, pages 475–482, Copenhagen.

Karthik Visweswariah, Jiri Navratil, Jeffrey Sorensen, Vijil Chenthamarakshan, and Nandakishore Kambhatla. 2010. Syntax based reordering with automatically derived rules for improved statistical machine translation. In *Proc. of Coling 2010*, pages 1119–1127.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proc.of EMNLP-CoNLL*, pages 737–745.

Xianchao Wu, Takuya Matsuzaki, and Jun'ichi Tsujii. 2010. Fine-grained tree-to-string translation rule extraction. In *Proc.of the 48th ACL*, pages 325–334.

Xianchao Wu, Takuya Matsuzaki, and Jun'ichi Tsujii. 2011. Effective use of function words for rule generalization in forest-based translation. In *Proc. of the 49th ACL-HLT*, pages 22–31.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proc.of COLING*, pages 508–514.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *Proc.of HLT-NAACL*, pages 245–253.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proc.of workshop on SSST*, pages 1–8.