Hierarchical Phrase-based Machine Translation with Word-based Reordering Model

Katsuhiko Hayashi*, Hajime Tsukada**
Katsuhito Sudoh, Kevin Duh**, Seiichi Yamamoto***
*Doshisha University
katsuhiko-h@is.naist.jp, seyamamo@mail.doshisha.ac.jp
**NTT Communication Science Laboratories
tsukada, sudoh, kevinduh@cslab.kecl.ntt.co.jp

Abstract

Hierarchical phrase-based machine translation can capture global reordering with synchronous context-free grammar, but has little ability to evaluate the correctness of word orderings during decoding. We propose a method to integrate word-based reordering model into hierarchical phrase-based machine translation to overcome this weakness. Our approach extends the synchronous context-free grammar rules of hierarchical phrase-based model to include reordered source strings, allowing efficient calculation of reordering model scores during decoding. Our experimental results on Japanese-to-English basic travel expression corpus showed that the BLEU scores obtained by our proposed system were better than those obtained by a standard hierarchical phrase-based machine translation system.

1 Introduction

Hierarchical phrase-based machine translation (Chiang, 2007; Watanabe et al., 2006) is one of the promising statistical machine translation approaches (Brown et al., 1993). Its model is formulated by a synchronous context-free grammar (SCFG) which captures the syntactic information between source and target languages. Although the model captures global reordering by SCFG, it does not explicitly introduce reordering model to constrain word order. In contrast, lexicalized reordering models (Tillman, 2004; Koehn et al., 2005; Nagata et al., 2006) are extensively used for phrase-based translation. These lexicalized reordering models cannot be directly applied to hierarchical phrased-based translation since the hierarchical phrase representation uses nonterminal symbols.

To handle global reordering in phrase-based translation, various preprocessing approaches have been proposed, where the source sentence is reordered to target language order beforehand (Xia and McCord, 2004; Collins et al., 2005; Li et al., 2007; Tromble and Eisner, 2009). However, preprocessing approaches cannot utilize other information in the translation model and target language model, which has been proven helpful in decoding.

This paper proposes a method that incorporates word-based reordering model into hierarchical phrase-based translation to constrain word order. In this paper, we adopt the reordering model originally proposed by Tromble and Eisner (2009) for the preprocessing approach in phrase-based translation. To integrate the word-based reordering model, we added a reordered source string into the right-hand-side of SCFG's rules. By this extension, our system can generate the reordered source sentence as well as target sentence and is able to efficiently calculate the score of the reordering model. Our method utilizes the translation model and target language model as well as the reordering model during decoding. This is an advantage of our method over the preprocessing approach.

The remainder of this paper is organized as follows. Section 2 describes the concept of our approach. Section 3 briefly reviews our proposed method on hierarchical phrase-based ma-

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 439–446,
Beijing, August 2010

Standard SCFG	$X \rightarrow< X1$ wa jinsei no $X2$ da , $X1$ is $X2$ of life$>$
SCFG (move-to-front)	$X \rightarrow< X1$ wa jinsei no $X2$ da , wa $X1$ da $X2$ no jinsei , $X1$ is $X2$ of life$>$
SCFG (attach)	$X \rightarrow< X1$ wa jinsei no $X2$ da , $X1$ wa da $X2$ no jinsei , $X1$ is $X2$ of life$>$

Table 1: A Japanese-to-English example of various SCFG's rule representations. Japanese words are romanized. Our proposed representation of rules has reordered source string to generate reordered source sentence $S^{'}$ as well as target sentence T. The "move-to-front" means Tromble and Eisner (2009) 's algorithm and the "attach" means Al-Onaizan and Papineni (2006) 's algorithm.

chine translation model. We experimentally compare our proposed system to a standard hierarchical phrase-based system on Japanese-to-English translation task in Section 4. Then we discuss on related work in Section 5 and conclude this paper in Section 6.

2 The Concept of Our Approach

The preprocessing approach (Xia and McCord, 2004; Collins et al., 2005; Li et al., 2007; Tromble and Eisner, 2009) splits translation procedure into two stages:

$$S \rightarrow S^{'} \rightarrow T \qquad (1)$$

where S is a source sentence, $S^{'}$ is a reordered source sentence with respect to the word order of target sentence T. Preprocessing approach has the very deterministic and hard decision in reordering. To overcome the problem, Li et al. (2007) proposed k-best appoach. However, even with a k-best approach, it is difficult to generate good hypotheses $S^{'}$ by using only a reordering model.

In this paper, we directly integrated the reordering model into the decoder in order to use the reordering model together with other information in the hierarchical phrase-based translation model and target language model. Our approach is expressed as the following equation.

$$S \rightarrow (S^{'}, T). \qquad (2)$$

Our proposed method generates the reordered source sentence $S^{'}$ by SCFG and evaluates the correctness of the reorderings using a word-based reordering model of S' which will be introduced in section 3.4.

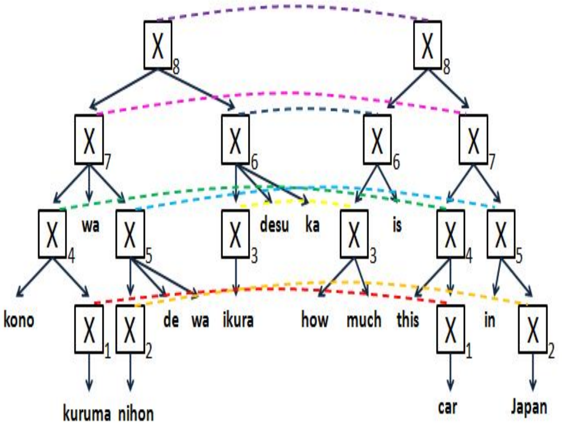

Figure 1: A derivation tree for Japanse-to-English translation.

3 Hierarchical Phrase-based Model Extension

3.1 Hierarchical Phrase-based Model

Hierarchical phrase-based model (Chiang, 2007) induces rules of the form

$$X \rightarrow < \gamma, \alpha, \sim, w > \qquad (3)$$

where X is a non-terminal symbol, γ is a sequence string of non-terminals and source terminals, α is a sequence string of non-terminals and target terminals. $\sim$ is a one-to-one correspondence for the non-terminals appeared in γ and α.

Given a source sentence S, the translation task under this model can be expressed as

$$\hat{T} = T\left(\underset{D:S(D)=S}{\text{argmax}}\ w(D)\right) \qquad (4)$$

where D is a derivation and $w(D)$ is a score of the derivation. Decoder seeks a target sentence

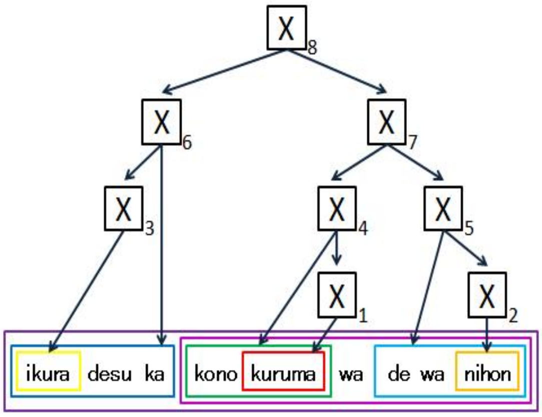

Figure 2: Reordered source sentence generated by our proposed system.

$T(D)$ which has the highest score $w(D)$. $S(D)$ is a source sentence under a derivation D. Figure 1 shows the example of Japanese-to-English translation by hierarchical phrase-based machine translation model.

3.2 Rule Extension

To generate reordered source sentence $S^{'}$ as well as target sentence T, we extend hierarchical phrase rule expressed in Equation 3 to

$$X \rightarrow < \gamma, \gamma^{'}, \alpha, \sim, w > \tag{5}$$

where $\gamma^{'}$ is a sequence string of non-terminals and source terminals, which is reordered γ with respect to the word order of target string α. The reason why we add $\gamma^{'}$ to rules is to efficiently calculate the reordering model scores. If each rule does not have $\gamma^{'}$, the decoder need to keep word alignments because we cannot know word order of $S^{'}$ without them. The calculation of reordering model scores using word alignments is very wasteful when decoding.

The translation task under our model extends Equation 4 to the following equation:

$$\hat{T} = (\hat{S}^{'}, \hat{T}) = (S^{'}, T)\left(\underset{D:S(D)=S}{\text{argmax}}\; w(D)\right). \tag{6}$$

Our system generates the reordered source sentence $S^{'}$ as well as target sentence T. Figure 2 shows the generated reordered source sentence $S^{'}$ when translating the example of Figure 1. Note that the structure of $S^{'}$ is the same as that of target sentence T. The decoder generates both Figure 2 and the right hand side of Figure 1, allowing us to score both global and local word reorderings.

Uni-gram Features
s_r, s-pos_r
s_r
s-pos_r
s_l, s-pos_l
s_l
s-pos_l

Bi-gram Features
s_r, s-pos_r, s_l, s-pos_l
s-pos_r, s_l, s-pos_l
s_r, s_l, s-pos_l
s_r, s-pos_r, s-pos_l
s_r, s-pos_r, s_l
s_r, s_l
s-pos_r, s-pos_l

Table 2: Features used by Word-based Reordering Model. pos means part-of-speech tag.

To add $\gamma^{'}$ to rules, we permuted γ into $\gamma^{'}$ after rule extraction based on Grow-diag-final (Koehn et al., 2005) alignment by GIZA++ (Och and Ney, 2003). To do this permutation on rules, we applied two methods. One is the same algorithm as Tromble and Eisner (2009), which reorders aligned source terminals and nonterminals in the same order as that of target side and moves unaligned source terminals to the front of aligned terminals or nonterminals (move-to-front). The other is the same algorithm as AI-Onaizan and Papineni (2006), which differs from Tromble and Eisner's approach in attaching unaligned source terminals to the closest prealigned source terminals or nonterminals (attach). This extension of adding $\gamma^{'}$ does not increase the number of rules.

Table 1 shows a Japanese-to-English example of the representation of rules for our proposed system. Japanese words are romanized. Suppose that source-side string is (X1 wa jinsei no X2 da) and target-side string is (X1 is X2 of life) and their word alignments are a=((jinsei , life) , (no , of) , (da , is)). Source-side aligned words and nonterminal symbols are sorted into the same order of target string. Source-side unaligned word (wa) is moved to the front or right of the prealigned symbol (X1).

Surrounding Word Pos Features
$s\text{-}pos_r$, $s\text{-}pos_r+1$, $s\text{-}pos_l-1$, $s\text{-}pos_l$
$s\text{-}pos_r-1$, $s\text{-}pos_r$, $s\text{-}pos_l-1$, $s\text{-}pos_l$
$s\text{-}pos_r$, $s\text{-}pos_r+1$, $s\text{-}pos_l$, $s\text{-}pos_l+1$
$s\text{-}pos_r-1$, $s\text{-}pos_r$, $s\text{-}pos_l$, $s\text{-}pos_l+1$

Table 3: The Example of Context Features

3.3 Word-based Reordering Model

We utilize the following $score(S^{'})$ as a feature for the word-based reordering model. This is incorporated into the log-linear model (Och and Ney, 2002) of statistical machine translation.

$$score(S^{'}) = \sum_{i,j:1\le i<j\le n} B[s^{'}_i, s^{'}_j] \tag{7}$$

$$B[s^{'}_l, s^{'}_r] = \theta \cdot \phi(s^{'}_l, s^{'}_r) \tag{8}$$

where n is the length of reordered source sentence $S^{'}$ $(= (s^{'}_1 \ldots s^{'}_n))$, θ is a weight vector and ϕ is a vector of features. This reordering model, which is originally proposed by Tromble and Eisner (2009), can assign a score to any possible permutation of source sentences. Intuitively $B[s^{'}_l, s^{'}_r]$ represents the score of ordering $s^{'}_l$ before $s^{'}_r$; the higher the value, the more we prefer word $s^{'}_l$ occurs before $s^{'}_r$. Whether $S^{'}_l$ should occur before $S^{'}_r$ depends on how often this reordering occurs when we reorder the source to target sentence order.

To train B, we used binary feature functions ϕ as used in (Tromble and Eisner, 2009), which were introduced for dependency parsing by McDonald et al. (2005). Table 2 shows the kind of features we used in our experiments. We did not use context features like surrounding word pos features in Table 3 because they were not useful in our preliminary experiments and propose an efficient implementation described in the next section in order to calculate this reordering model when decoding. To train the parameter θ, we used the perceptron algorithm following Tromble and Eisner (2009).

3.4 Integration to Cube Pruning

CKY parsing and cube-pruning are used for decoding of hierarchical phrase-based model (Chiang, 2007). Figure 3 displays that hierarchical phrase-based decoder seeks new span [1,7] items with rules, utilizing subspan [1,3] items and subspan [4,7] items. In this example, we use 2-gram language model and +LM decoding. uni(•) means 1-gram language model cost for heuristics and interaction usually means language model cost that cannot be calculated offline. Here, we introduce our two implementations to calculate word-based reordering model scores in this decoding algorithm.

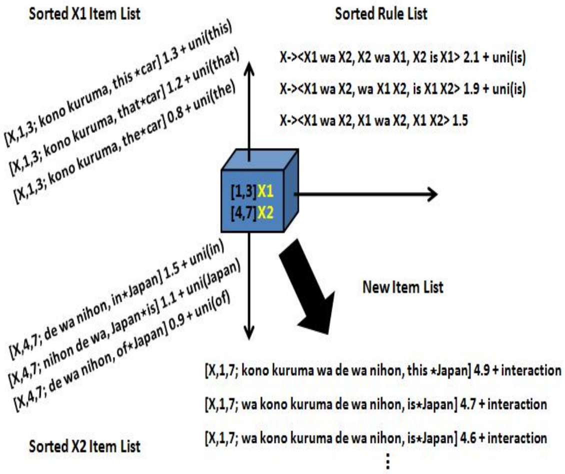
Sorted X1 Item List
Sorted Rule List
[X,1,3; kono kuruma, this∗car] 1.3 + uni(this)
[X,1,3; kono kuruma, that∗car] 1.2 + uni(that)
[X,1,3; kono kuruma, the∗car] 0.8 + uni(the)
X-><X1 wa X2, X2 wa X1, X2 is X1> 2.1 + uni(is)
X-><X1 wa X2, wa X1 X2, is X1 X2> 1.9 + uni(is)
X-><X1 wa X2, X1 wa X2, X1 X2> 1.5
[1,3] X1
[4,7] X2
[X,4,7; de wa nihon, in∗Japan] 1.5 + uni(in)
[X,4,7; nihon de wa, Japan∗is] 1.1 + uni(Japan)
[X,4,7; de wa nihon, of∗Japan] 0.9 + uni(of)
New Item List
[X,1,7; kono kuruma wa de wa nihon, this ∗Japan] 4.9 + interaction
[X,1,7; wa kono kuruma de wa nihon, is∗Japan] 4.7 + interaction
[X,1,7; wa kono kuruma de wa nihon, is∗Japan] 4.6 + interaction
⋮
Sorted X2 Item List

Figure 3: Creating new items from subitems and rules, that have a span [1,7] in source sentence.

First, we explain a naive implementation shown in the left side of Figure 4. This algorithm performs the same calculation of reordering model as that of language model. Each item keeps a part of reordered source sentence. The reordering score of new item can be calculated as interaction cost when combining subitems with the rule.

The right side of Figure 4 shows our proposed implementation. This implementation can be adopted to decoding only when we do not use context features like surrounding word pos features in Table 3 (and consider a distance between words in features). If a span is given, the reordering scores of new item can be calculated for each rule, being independent from the word order of reordered source segment of a subitem. So, the reordering model scores can be calculated for all rules with spans by using a part of the input source sentence before sorting them for cube pruning. We expect this sorting of rules with reordering

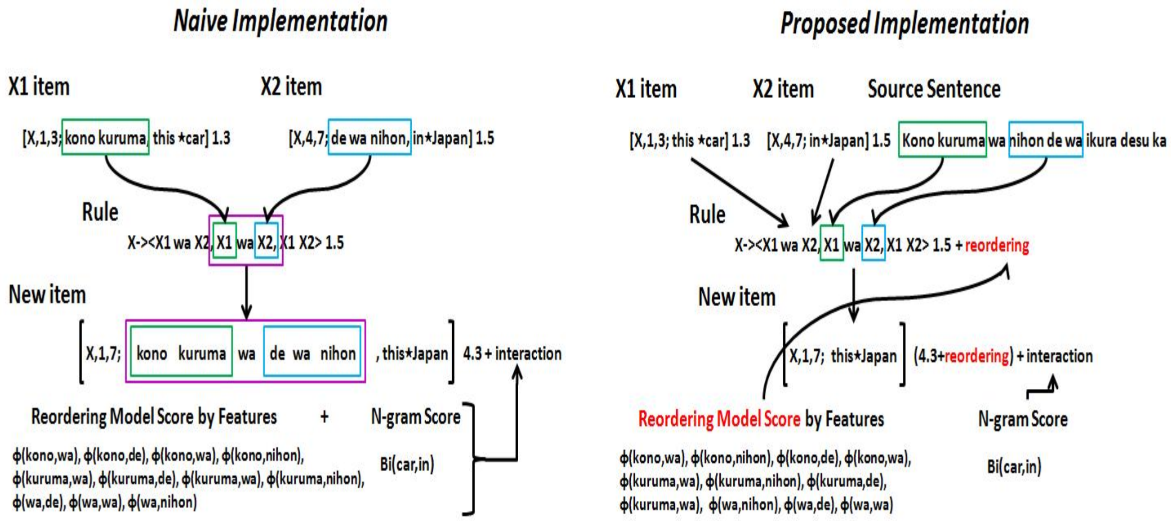
Naive Implementation
X1 item
X2 item
[X,1,3; kono kuruma, this ⋆car] 1.3
[X,4,7; de wa nihon, in⋆Japan] 1.5
Rule
X-><X1 wa X2, X1 wa X2, X1 X2> 1.5
New item
X,1,7; kono kuruma wa de wa nihon , this⋆Japan
4.3 + interaction
Reordering Model Score by Features
+
N-gram Score
ϕ(kono,wa), ϕ(kono,de), ϕ(kono,wa), ϕ(kono,nihon), ϕ(kuruma,wa), ϕ(kuruma,de), ϕ(kuruma,wa), ϕ(kuruma,nihon), ϕ(wa,de), ϕ(wa,wa), ϕ(wa,nihon)
Bi(car,in)
Proposed Implementation
X1 item
X2 item
Source Sentence
[X,1,3; this ⋆car] 1.3
[X,4,7; in⋆Japan] 1.5
Kono kuruma wa nihon de wa ikura desu ka
Rule
X-><X1 wa X2, X1 wa X2, X1 X2> 1.5 + reordering
New item
X,1,7; this⋆Japan
(4.3+reordering) + interaction
Reordering Model Score by Features
N-gram Score
ϕ(kono,wa), ϕ(kono,nihon), ϕ(kono,de), ϕ(kono,wa), ϕ(kuruma,wa), ϕ(kuruma,nihon), ϕ(kuruma,de), ϕ(kuruma,wa), ϕ(wa,nihon), ϕ(wa,de), ϕ(wa,wa)
Bi(car,in)

Figure 4: The “naive” and “proposed” implementation to calculate the reordering cost of new items.

model scores will have good influence on cube pruning. The right hand side of Figure 4 shows the diffrence between naive and proposed implementation ($S^{'}$ is not shown to allow for a clear presentation). Note the difference is in where/when the reordering scores are inserted: together with the N-gram scores in the case of naive implementation; incorpolated into sorted rules for the proposed implementation.

4 Experiment

4.1 Purpose

To reveal the effectiveness of integrating the reordering model into decoder, we compared the following setups:

- baseline: a standard hierarchical phrase-based machine translation (Hiero) system.

- preprocessing: applied Tromble and Eisner’s approach, then translate by Hiero system.

- Hiero system + reordering model: integrated reordering model into Hiero system.

We used the Joshua Decoder (Li and Khudanpur, 2008) as the baseline Hiero system. This decoder uses a log-linear model with seven features, which consist of N-gram language model $P_{LM}(T)$, lexical translation model $P_w(\gamma|\alpha)$, $P_w(\alpha|\gamma)$, rule translation model $P(\gamma|\alpha)$, $P(\alpha|\gamma)$, word penalty and arity penalty.

The “Hiero + Reordering model” system has word-based reordering model as an additional feature to baseline features. For this approach, we use two systems. One has “move-to-front” system and the other is “attach” system explained in Section 3.2. We implemented our proposed algorithm in Section 3.4 to both “Hiero + Reordering model” systems. As for beam width, we use the same setups for each system.

4.2 Data Set

Data		Sent.	Word.	Avg. leng
Training	ja	200.8K	2.4M	12.0
	en	200.8K	2.3M	11.5
Development	ja	1.0K	10.3K	10.3
	en	1.0K	9.8K	9.8
Test	ja	1.0K	14.2K	14.2
	en	1.0K	13.5K	13.5

Table 4: The Data statistics

For experiments we used a Japanese-English basic travel expression corpus (BTEC). Japanese word order is linguistically very different from English and we think Japanese-English pair is a very good test bed for evaluating reordering model.

Metrics System	BLEU	PER
Baseline (Hiero)	28.09	39.68
Preprocessing	17.32	45.27
Hiero + move-to-front	**28.85**	39.89
Hiero + attach	**29.25**	**39.43**

Table 5: BLEU and PER scores on the test set.

Our training corpus contains about 200.8k sentences. Using the training corpus, we extracted hierarchical phrase rules and trained 4-gram language model and word-based reordering model. Parameters were tuned over 1.0k sentences (development data) with single reference by minimum error rate training (MERT) (Och, 2003). Test data consisted of 1.0k sentences with single reference. Table 4 shows the condition of corpus in detail.

4.3 Results

Table 5 shows the BLEU (Papineni et al., 2001) and PER (Niesen et al., 2000) scores obtained by each system. The results clearly indicated that our proposed system with word-based reordering model (move-to-front or attach) outperformed baseline system on BLEU scores. In contrast, there is no significant improvement from baseline on PER. This suggests that the improvement of BLEU mainly comes from reordering. In our experiment, preprocessing approach resulted in very poor scores.

4.4 Discussion

Table 6 displays examples showing the cause of the improvements of our system with reordering model (attach) comparing to baseline system. We can see that the outputs of our system are more fluent than those of baseline system because of reordering model.

As a further analysis, we calculated the BLEU scores of Japanese $S^{'}$ predicted from reordering model against true Japanese $S^{'}$ made from GIZA++ alignments, were only 26.2 points on development data. We think the poorness mainly comes from unaligned words since they are untractable for the word-based reordering model. Actually, Japanese sentences in our training data include 34.7% unaligned words. In spite of the poorness, our proposed method effectively utilize this reordering model in contrast to preprocessing approach.

5 Related Work

Our approach is similar to preprocessing approach (Xia and McCord, 2004; Collins et al., 2005; Li et al., 2007; Tromble and Eisner, 2009) in that it reorders source sentence in target order. The difference is this sentence reordering is done in decoding rather than in preprocessing.

A lot of studies on lexicalized reordering (Tillman, 2004; Koehn et al., 2005; Nagata et al., 2006) focus on the phrase-based model. These works cannnot be directly applied to hierarchical phrase-based model because of the difference between normal phrases and hierarchical phrases that includes nonterminal symbols.

Shen et al. (2008,2009) proposed a way to integrate dependency structure into target and source side string on hierarchical phrase rules. This approach is similar to our approach in extending the formalism of rules on hierarchical phrase-based model in order to consider the constraint of word order. But, our approach differs from (Shen et al., 2008; Shen et al., 2009) in that syntax annotation is not necessary.

6 Conclusion and Future Work

We proposed a method to integrate word-based reordering model into hierarchical phrase-based machine translation system. We add $\gamma^{'}$ into the hiero rules, but this does not increase the number of rules. So, this extension itself does not affect the search space of decoding. In this paper we used Tromble and Eisner's reordering model for our method, but various reordering model can be incorporated to our method, for example $S^{'}$ N-gram language model. Our experimental results on Japanese-to-English task showed that our system outperformed baseline system and preprocessing approach.

In this paper we utilize $\gamma^{'}$ only for reordering model. However, it is possible to use $\gamma^{'}$ for other modeling, for example we can use it for rule translation probabilities $P(\gamma^{'}|\gamma)$, $P(\gamma|\gamma^{'})$ for additional feature functions. Of course, we can

S	america de seihin no hanbai wo hajimeru keikaku ga ari masu ka .	kono tegami wa koukuubin de nihon made ikura kakari masu ka .
T_B	sales of product in america are you planning to start ?	this letter by airmail to japan . how much is it ?
T_P	are you planning to start products in the u.s. ?	how much does it cost to this letter by airmail to japan ?
R	do you plan to begin selling your products in the u.s. ?	how much will it cost to send this letter by air mail to japan ?

Table 6: Examples of outputs for input sentence S from baseline system T_B and our proposed system (attach) T_P. R is a reference. The underlined portions have equivalent meanings and show the reordering differences.

also utilize reordered target sentence $T^{'}$ for various modeling as well. Addtionally we plan to use $S^{'}$ for MERT because we hypothesize the fluent $S^{'}$ leads to fluent T.

References

AI-Onaizan, Y. and K. Papineni. 2006. Distortion models for statistical machine translation. In *Proc. the 44th ACL*, pages 529–536.

Brown, P. F., S. A. D. Pietra, V. D. J. Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguitics*, 19:263–312.

Chiang, D., K. Knight, and W. Wang. 2009. 11,001 new features for statistical machine translation. In *Proc. NAACL*, pages 216–226.

Chiang, D. 2007. Hierachical phrase-based translation. *Computational Linguitics*, 33:201–228.

Collins, M., P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. the 43th ACL*, pages 531–540.

Collins, M. 2002. Discriminative training methods for hidden markov models. In *Proc. of EMNLP*.

Freund, Y. and R. E. Schapire. 1996. Experiments with a new boosting algorithm. In *Proc. of the 13th ICML*, pages 148–156.

Koehn, P., A. Axelrod, A-B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot. 2005. Edinburgh system description for 2005 iwslt speech translation evaluation. In *Proc. the 2nd IWSLT*.

Li, Z. and S. Khudanpur. 2008. A scalable decoder for parsing-based machine translation with equivalent language model state maintenance. In *Proc. ACL SSST*.

Li, C-H., D. Zhang, M. Li, M. Zhou, K. Li, and Y. Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proc. the 45th ACL*, pages 720–727.

McDonald, R., K. Crammer, and F. Pereira. 2005. Spanning tree methods for discriminative training of dependency parsers. In *Thechnical Report MS-CIS-05-11, UPenn CIS*.

Nagata, M., K. Saito, K. Yamamoto, and K. Ohashi. 2006. A clustered global phrase reordering model for statistical machine translation. In *COLING-ACL*, pages 713–720.

Niesen, S., F.J. Och, G. Leusch, and H. Ney. 2000. An evaluation tool for machine translation: Fast evaluation for mt research. In *Proc. the 2nd International Conference on Language Resources and Evaluation*.

Och, F. J. and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. the 40th ACL*, pages 295–302.

Och, F. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.

Och, F. J. 2003. Minimum error rate training in statistical machine translation. In *Proc. the 41th ACL*, pages 160–167.

Papineni, K. A., S. Roukos, T. Ward, and W-J. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. In *Proc. the 39th ACL*, pages 311–318.

Shen, L., J. Xu, and R. Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. ACL*, pages 577–585.

Shen, L., J. Xu, B. Zhang, S. Matsoukas, and R. Weischedel. 2009. Effective use of linguistic and contextual information for statistical machine translation. In *Proc. EMNLP*, pages 72–80.

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

Tromble, R. and J. Eisner. 2009. Learning linear ordering problems for better translation. In *Proc. EMNLP*, pages 1007–1016.

Watanabe, T., H. Tsukada, and H. Isozaki. 2006. Left-to-right target generation for hierarchical phrase-based translation. In *Proc. COLING-ACL*, pages 777–784.

Xia, F. and M. McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proc. the 18th ICON*, pages 508–514.