HSSA Tree Structures for BTG-based Preordering in Machine Translation

Yujia Zhang[1,2], **Hao Wang**[1] **and Yves Lepage**[1]
[1]Graduate School of Information, Production and Systems
Waseda University, Kitakyushu, Fukuoka 808-0135, Japan
[2]School of Computer Engineering and Science,
Shanghai University, Shanghai 200444, China
{ashley.zhang@moegi., oko_ips@ruri., yves.lepage@}waseda.jp

Abstract

The Hierarchical Sub-Sentential Alignment (HSSA) method is a method to obtain aligned binary tree structures for two aligned sentences in translation correspondence. We propose to use the binary aligned tree structures delivered by this method as training data for preordering prior to machine translation. For that, we learn a Bracketing Transduction Grammar (BTG) from these binary aligned tree structures. In two oracle experiments in English to Japanese and Japanese to English translation, we show that it is theoretically possible to outperform a baseline system with a default distortion limit of 6, by about 2.5 and 5 BLEU points and, 7 and 10 RIBES points respectively, when preordering the source sentences using the learnt preordering model and using a distortion limit of 0. An attempt at learning a preordering model and its results are also reported.

1 Introduction

One of the major common challenges for machine translation (MT) is the different order of the same conceptual units in the source and target languages. In order to get a fluent and adequate translation in the target language, the default phrase-based statistical machine translation (PB-SMT) system implemented in MOSES has a simple distortion model using position (Koehn et al., 2003) and lexical information (Tillmann, 2004) to allow reordering during decoding. Other solutions exist: e.g., the distortion model in (Al-Onaizan and Papineni, 2006) handles n-gram language model limitations; Setiawan et al. (2007) propose a function word centered syntax-based (FWS) solution; Zhang et al. (2007) propose a reordering model integrating syntactic knowledge. Also, other models than the phrase-based model have been proposed to address the reordering problem, like hierarchical phrase-based SMT (Chiang, 2007) or syntax-based SMT (Yamada and Knight, 2001).

Preordering (Xia and McCord, 2004; Collins et al., 2005) has been proposed primarily to solve the problems encountered when translating between languages with widely divergent syntax, for instance, from a subject-verb-object (SVO) language (like English and Mandarin Chinese) to a subject-object-verb (SOV) language (like Japanese and Korean), Preordering is a pre-processing task that aims to rearrange the word order of a source sentence to fit the word order of the target language. It is separated from the core translation task. Recent approaches (DeNero and Uszkoreit, 2011; Neubig et al., 2012; Nakagawa, 2015) learn a preordering model based on Bracketing Transduction Grammar (BTG) (Wu, 1997) from parallel texts to score permutations by using tree structures as latent variables. They build the needed tree structures and the preordering model (i.e., a BTG) at the same time using word alignments. However it is needed to check whether a given sentence can fit the desired tree structures.

It seems of course more difficult to build both the tree structures and the preordering model at the same time than to build only a preordering model if the tree structures are given. In this paper, we rapidly obtain tree structures using word-to-word associations taking advantage of the hierarchical sub-sentential alignment (HSSA) method (Lardilleux et al., 2012). This method computes a recursive binary segmentation in both languages at the same

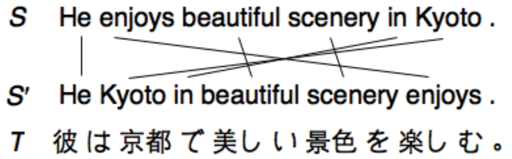
S He enjoys beautiful scenery in Kyoto .
S′ He Kyoto in beautiful scenery enjoys .
T 彼 は 京都 で 美しい 景色 を 楽しむ 。

Figure 1: Example of preordering.

time, judging whether two spans with the same concepts in both languages are inverted or not. We conduct oracle experiments to show that these tree structures may be beneficial for PB-SMT. We then use these tree structures as the training data to build a preordering model without checking the validity by modifying the top-down BTG parsing method introduced in (Nakagawa, 2015). Oracle experiments show that if we reorder source sentences exactly, translation scores can be improved by around 2.5 BLEU points and 7 RIBES points in English to Japanese) and 5 BLEU points and 10 RIBES points in Japanese to English. Experiments with our tree structures show that better RIBES scores can be easily obtained.

The rest of this paper is organized as follows: Section 2 describes related work in preordering and BTG-based preordering. Section 3 shows how to obtain tree structures using word-to-word associations. Section 4 reports oracle preordering experiments. Section 5 gives a method to build a preordering model using tree structures. Section 6 presents the results of our experiments and their analysis.

2 Related Work

2.1 Preordering for SMT

Preordering in statistical machine translation (SMT) converts a source sentence S, before translation, into a reordered source sentence S', where the word order is similar to that of the target sentence T (Figure 1).

Preordering can be seen as an optimization problem, where we want to find the best reordered source sentence that maximizes the probability among all possible reordering of the sentence.

$$\hat{S'} = \underset{S' \in \gamma(S)}{\operatorname{argmax}} P(S'|S) \tag{1}$$

$\hat{S'}$ represents the best reordered source sentence, and $\gamma(S)$ stands for the set of all possible reordering of the source sentence.

Syntax-based preordering based on the existence parsers has been proposed to pre-process the source sentences by using automatically learned rewriting patterns (Xia and McCord, 2004). Several methods have been proposed methods, such as constituent parsing by automatically extracting preordering rules from a parallel corpus (Xia and McCord, 2004; Wu et al., 2011) or by creating rules manually (Wang et al., 2007; Han et al., 2012), or dependency parsing with automatically created rules (Habash, 2012; Cai et al., 2014) or manually generated rules (Xu et al., 2009; Isozaki et al., 2010).

Another trend of research is to try to solve the preordering problem without relying on parsers. Tromble and Eisner (2009) propose sophisticated reordering models based on the Linear Ordering Problem. Visweswariah et al. (2011) learn a preordering model by similarity with the Traveling Salesman Problem. Lerner and Petrovs (2013) present a source-side classifier-based preordering model. Several pieces of research (DeNero and Uszkoreit, 2011; Neubig et al., 2012; Nakagawa, 2015) are mainly about using tree structures as latent variables for preordering models. This is detailed in the next subsection.

2.2 BTG-based Preordering

BTG-based preordering is based on Bracketing Transduction Grammar (BTG), also called Inversion Transduction Grammar (ITG) (Wu, 1997). Whereas Chomsky Normal Form of context-free rules has two types of rules ($X \rightarrow X_1X_2$ and $X \rightarrow x$) and the grammar is monolingual, BTG has three types of rules, Straight, Inverted and Terminal, to cope with the possible correspondences between a source language and a target language.

Straight keeps the same order in the source and the target languages; Inverted exchanges the order; Terminal just stands for the production of a non-terminal symbol both in the source and target languages. The corresponding tree structures are illustrated in Figure 3 from (a) to (c) in the same order. The parse tree obtained by applying a BTG to parse a pair of sentences, provides the necessary information to reorder the source sentence in conformity to the word order of the target sentence, as it suffices to

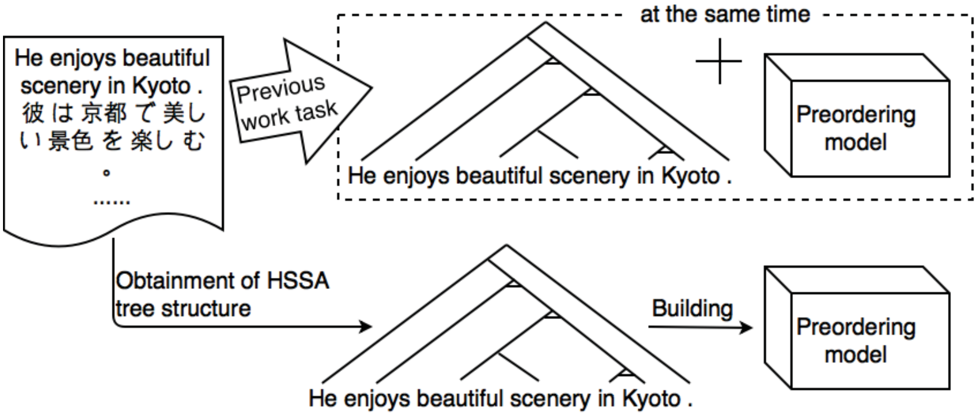
He enjoys beautiful scenery in Kyoto .
彼 は 京都 で 美し い 景色 を 楽し む 。
......
Previous work task
at the same time
He enjoys beautiful scenery in Kyoto .
+
Preordering model
Obtainment of HSSA tree structure
He enjoys beautiful scenery in Kyoto .
Building
Preordering model

Figure 2: The difference between previous methods (Neubig et al., 2012; Nakagawa, 2015) and our proposed method when building a preordering model. In previous work, the tree structures and the preordering model should be deduced at the same time from the parallel text. Our work firstly produces the tree structures from parallel text, and then computes a preordering model.

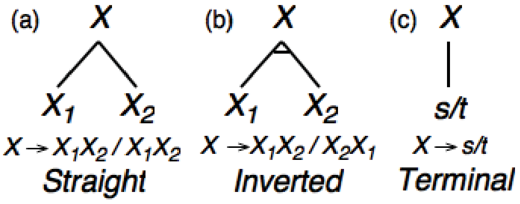

Figure 3: Tree structures related to bracketing transduction grammar.

read the type of rules applied, straight or inverted.

Neubig et al. (2012) present a discriminative parser using the derivations of tree structures as underlying variables from word alignment with the parallel corpus. However, the computation complexity is $O(n^5)$ for a sentence length of n because the method guesses the tree structure using the Coke-Younger-Kasami (CYK) algorithm, which complexity is $O(n^3)$. In order to reduce complexity, Nakagawa (2015) proposes a top-down BTG parsing approach instead of the bottom-up CYK algorithm. The computation complexity reduces to $O(kn^2)$ for a sentence length of n and a beam width of k.

Both methods need to predict the possible tree structures for each sentence when building the preordering model. Word alignments are used to check whether a pair of sentences can yield a valid tree structure.[1] Predicting tree structures while building the preordering model at the same time is difficult. In the present paper, we propose to directly generate the tree structures from the word-to-word association matrices, and to use these tree structures to build the preordering model afterwards. Figure 2 illustrates the differences between the two previous methods and our proposed method.

3 Obtaining HSSA Tree Structures

In our proposed method, the tree structures are obtained by using soft alignment matrices and recursively segmenting these matrices with Ncut scores (Zha et al., 2001) using the hierarchical sub-sentential alignment (HSSA) method (Lardilleux et al., 2012).

The HSSA method delivers tree structures which are similar to parse trees obtained by the application of a BTG. Figure 4 shows that segmenting along the second diagonal with the HSSA method corresponds to an Inverted rule in the BTG formalism and that segmenting according to the first diagonal corresponds to Straight. The column $S_p.\overline{S_p}$[2] and the row $T_p.\overline{T_p}$ of the matrix in Figure 4 are related to part of the source sentence and part of the target sentence respectively.

The HSSA method uses soft alignment matrices

[1] A sentence pair which cannot be represented by a BTG tree structure is: $B_2D_4A_1C_3$ to $A_1B_2C_3D_4$.

[2] The symbol "." stands for the concatenation of word strings.

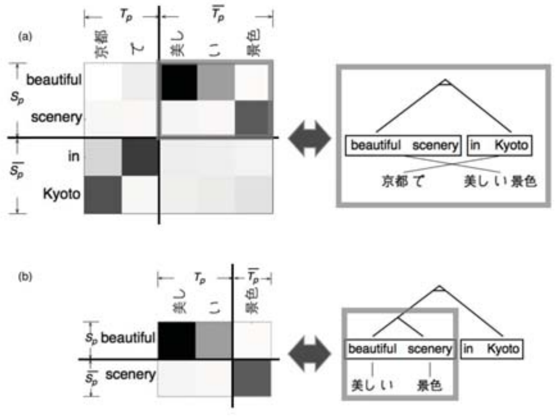

Figure 4: Hierarchical sub-sentential alignment and generation of tree structures. (a) a best segmentation according to the second diagonal in the soft alignment matrix using the HSSA method coresponds to an Inverted rule in the BTG formalism; (b) a best segmentation according to the main diagonal corresponds to a Straight rule. (b) is a sub-part in (a) to illustrate recursivity.

where each cell for a source word s and a target word t has a score $w(s,t)$ computed as the geometric mean of the word-to-word translation probabilities in both directions (see Equation (2)). In Figure 4, the saturation of the cells represents the score $w(s,t)$: the darker the color, the higher the score.

$$w(s,t) = \sqrt{p(s|t) \times p(t|s)} \tag{2}$$

Each segmentation iteration segments the soft alignment matrix in both horizontal and vertical directions to decompose the matrix recursively into two corresponding sub-parts. There are two cases: the two sub-parts follow the main diagonal, (S_p, T_p) and $(\overline{S_p}, \overline{T_p})$, this is similar to the BTG rule Straight (see Figure 4(b)); or they follow the second diagonal, $(S_p, \overline{T_p})$ and $(\overline{S_p}, T_p)$, this is similar to the BTG rule Inverted (see Figure 4(a)). In order to decide for the segmentation point and for the direction in a submatrix $(X,\ Y) \in \{S_p,\ \overline{S_p}\} \times \{T_p,\ \overline{T_p}\}$, Ncut scores (Zha et al., 2001) of crossing points in the matrix $(S_p.\overline{S_p},\ T_p.\overline{T_p})$ are calculated in both directions.

$$W(X,Y) = \sum_{s \in X, t \in Y} w(s,t) \tag{3}$$

$$\text{cut}(X,Y) = W(X,\overline{Y}) + W(\overline{X},Y) \tag{4}$$

$$\begin{aligned} \text{Ncut}(X,Y) = \; & \frac{\text{cut}(X,Y)}{\text{Ncut}(X,Y) + 2 \times W(X,Y)} \\ & + \frac{\text{cut}(\overline{X},\overline{Y})}{\text{Ncut}(\overline{X},\overline{Y}) + 2 \times W(\overline{X},\overline{Y})} \end{aligned} \tag{5}$$

One tree structure for one sentence is generated with sub-sentential alignments at the same time by remembering the best segmentation point of each iteration in a sentence, using the HSSA method. In our proposed method, all the tree structures obtained from a training bilingual corpus become a training data set to learn a preordering model. The HSSA approach allows to get tree structures easily and rapidly, by using only a parallel corpus and the word-to-word associations obtained from it. No further annotation is needed.

4 Oracle Experiments: Upper Bounds

So as to check whether our proposed method is promising, in a first step, we perform oracle experiments. The purpose is to determine the upper bounds that can be obtained in translation evaluation scores. This will offer a judgment on the theoretical effectiveness of utilizing tree structures generated by the hierarchical sub-sentential alignment method.

In the oracle experiments, we apply the HSSA method on the sentence pairs of the test set to obtain their tree structures and then use these tree structures to reorder the source sentences of the test set. In a real experiment, this is impossible, because the target sentence, and hence the soft alignment matrices are unknown.

To reorder the words in a source sentence, as explained above, we recursively traverse the tree structure in a top-down manner. The order of the words in the source sentence is changed according to the types of nodes encountered in the tree structures. When the type of node is Straight, the two spans in the source sentence keep the original order; when it is Inverted, the two spans in the source sentence are inverted. After reordering, the alignment between the reordered source sentence and the target sentence follows the main diagonal, up to the cases where one word corresponds to several words. Figure 5 shows an example.

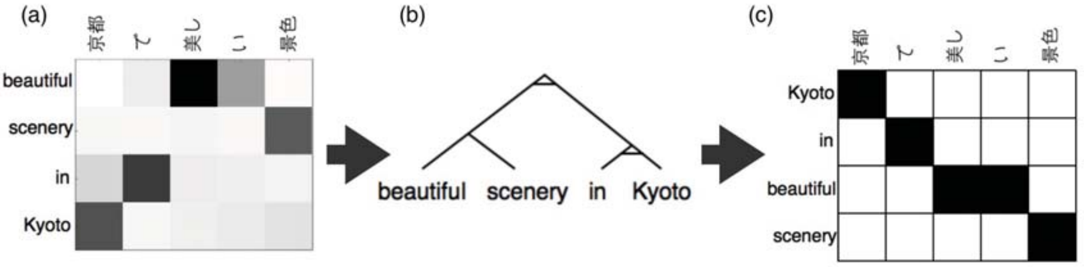

Figure 5: Example for oracle experiment. (a) a soft alignment matrix between a source sentence (left) and a target sentence (above); (b) a tree structure with Straight or Inverted nodes; (c) the alignment between the reordered source sentence and the target sentence. The arrow from (a) to (b) represents the generation of tree structures from word-to-word associations by use of the HSSA method; the arrow from (b) to (c) is reordering. In the oracle experiment, this is applied on test data. In a real experiment, this is applied on test data and development data, while the scheme given in Figure 6 is applied on the test data.

After reordering all source sentences in the training, tuning, and test sets, a standard PB-SMT system is built as usual with the reordered source sentences in place of the original sources sentences, and with their corresponding target sentences.

5 Building and Applying a Preordering Model

A preordering model is built by using the tree structures obtained on the parallel corpus used as training data for machine translation, as its training data. On test data, i.e., source sentences alone, the role of the pre-ordering model is to guess a new order for the words of the source sentences in the absence of corresponding target sentences. Figure 6 illustrates the process of building the preordering model with the tree structures obtained as explained in Figure 1 from the sentence pairs of the training data of a machine translation system. We now present a method to learn and apply a preordering model. This method is a modification of the top-down BTG parsing method presented in (Nakagawa, 2015). The main difference is that, in our present configuration, tree structures are available from a parallel corpus.

In Nakagawa's method, word alignments are used to predict the tree structures, so that, after segmenting one span into two, whether a word in one of two spans aligns to another word in the other span is checked in each iteration. However, in our configuration, we are able to directly get the separating points because we know the tree structure produced by the HSSA method.

The best derivation $\hat{d}$ for a sentence is important for both learning and applying a preordering model. Because one derivation leads to one parse tree, finding the best derivation can be regarded as finding the best parse tree. To assess the quality of a parse tree, we compare it with the tree structure output by the HSSA method. The best parse tree is the tree with the maximal score defined by the following formula:

$$\hat{d} = \underset{d \in D(T)}{\text{argmax}} \sum_{m \in \text{Nodes}(T)} \sigma(m) \quad (6)$$

where d represents one derivation in the set of all possible derivations $D(T)$ for the tree structure T; m represents one node in the set of nodes $\text{Nodes}(T)$ of the tree structure T, and $\sigma(m)$ represents the score of the node.

The score of a node in a tree structure is computed by applying the perceptron algorithm (Collins and Roark, 2004), i.e., by taking each node of trees as a latent variable (Nakagawa, 2015). This algorithm is an online learning algorithm, and processes nodes in an available tree structure one by one, by using the following formula to calculate the score of each node $\sigma(m)$:

$$\sigma(m) = \Lambda \cdot \Phi(m), \quad m \in \text{Nodes}(T)$$

where $\Phi(m)$ represents the feature vector of this node, and Λ represents the vector of feature weights.

Due to iterated binary decomposition, an increasing number of iterations for one sentence results in many derivations that wait for being checked

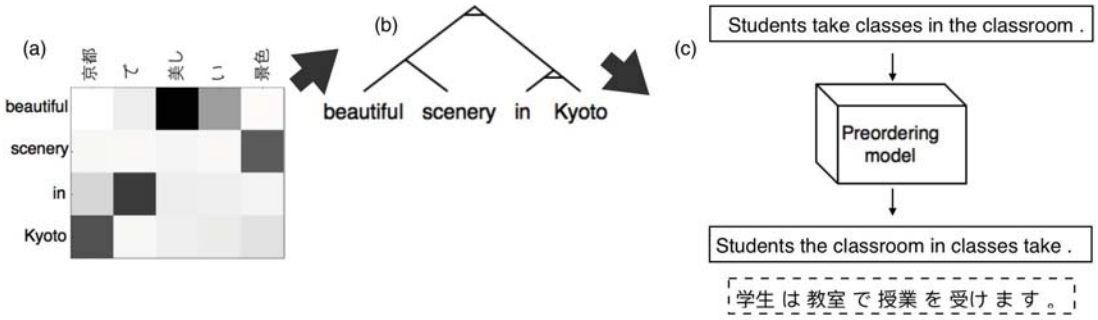
(a)
京都
で
美し
い
景色
beautiful
scenery
in
Kyoto
(b)
beautiful scenery in Kyoto
(c)
Students take classes in the classroom .
Preordering model
Students the classroom in classes take .
学生 は 教室 で 授業 を 受け ます 。

Figure 6: Example of building and applying preordering model using tree structures as the reference. (a), (b) and the arrow from (a) to (b) are the same with Figure 5. The difference is that both (a) and (b) generating from only a training set. (c) a sentence from test set becomes a target-like source sentence in the solid line and in dotted line it shows corresponding target sentence. The arrow from (b) to (c) represents building preordering model.

whether they are the best ones or not, both while building and while applying the preordering model. In order to control the size of the search space, a beam search is used.

We need to enable the system to output $\hat{d}$ to become as similar as possible as the derivation d found in the tree structure obtained by the HSSA model while building the preordering model. To do so, we learn the feature vectors and adjust their weight vectors by using the Expectation–Maximization (EM) algorithm on the training data. In the end, we obtain a preordering model with features and corresponding weights.

We then apply the preordering model on all the source sentences of all three data sets, training, tuning, and test, to reorder their words. A standard PB-SMT system is then built as usual with reordered source sentences in place of the original sources sentences, and with their corresponding target sentences.

6 Experiments

6.1 Experimental Settings

We build our PB-SMT systems in a standard way using the Moses system (Koehn et al., 2007), KenLM for language modelling (Heafield, 2011), and standard lexical reordering model (Koehn et al., 2005). This lexical reordering model allows local reordering with a given distortion limit during decoding. The default of the distortion limit in Moses is 6. When set to 0, the system does not perform any lexical reordering.

	Sentence	Words	
	Pairs	Japanese	English
Train	330,000	6.09 M	5.91 M
Tune	1,235	34.4 k	30.8 k
Test	1,160	28.5 k	26.7 k

Table 1: Number of sentences and words in the training, tuning and test sets of the KFTT corpus.

The language pair we work on is Japanese–English in both directions. The data sets are the training, tuning and test sets from the Kyoto Free Translation Task (KFTT) corpus.[3] In this corpus, Japanese sentences have been segmented and tokenized by KyTea.[4] Table 1 gives statistics on these data sets.

For the generation of tree structures, word-to-word associations are extracted from the training set andused to the hierarchical sub-sentential alignment method, are extracted only from the training set.

For our preordering model, we carried out experiments by following the experimental settings reported in (Nakagawa, 2015) with a beam search of 20, a number of iteration of 20 and 100,000 sentences pairs as preordering training extracted at random from the training set. We use three kinds of features, LEX, POS, and CLASS. LEX consists in the lexical items inside a given window around the current word in the source language. POS are the parts-of-speech of the lexical items of the LEX fea-

[3] http://www.phontron.com/kftt/index.html
[4] http://www.phontron.com/kytea/

ture words. The CLASS features are their semantic classes. The POS tagging information is provided by KyTea for Japanese, and the Lookahead Part-Of-Speech Tagger (Tsuruoka et al., 2011) for English.[5] We use the Brown clustering algorithm (Brown et al., 1992; Liang, 2005) for word class information in English and Japanese.

6.2 Evaluation Metrics

In order to evaluate the efficiency of reordering, we use a modified version of the Fuzzy Reordering Score (FRS) (Talbot et al., 2011) and Kendall's τ (Kendall, 1938) as intrinsic evaluation metrics. The modified version of FRS (see Equation (7)) is inspired by (Nakagawa, 2015) because only two words are considered and the indices of the first and the last words are also considered (Neubig et al., 2012).

$$\text{mod FRS} = \frac{B}{|S| + 1} \tag{7}$$

B represents the number of word bigrams which appear in both the reordered sentence and the golden reference, and $|S|$ represents the length of the source sentence S in words.

We also change the formula for calculating Kendall's τ to a normalized Kendall's τ following (Isozaki et al., 2010). Equation (8) gives the definition.

$$\text{norm } \tau = 1 - \frac{E}{|S| \times (|S| - 1)/2} \tag{8}$$

E represents the number of not increasing word pairs and $|S| \times (|S| - 1)/2$ is the total number of pairs.

Being a metric to evaluate the quality of machine translation, RIBES (Isozaki et al., 2010) is an extrinsic metric in our work. However, given the fact that RIBES takes order into account, it can also be considered an intrinsic metric in our work. As a matter of fact, RIBES bases on the computation of FRS and τ.

In addition, we of course use BLEU (Papineni et al., 2002) for the evaluation of machine translation quality as it is the de facto standard metric.

6.3 Experimental Results and Analysis

Table 2 shows the evaluation results in all intrinsic evaluation metrics (modified FRS and normalized τ), the intrinsic and extrinsic evaluation metric (RIBES) and in the extrinsic evaluation metric (BLEU). We use all these metrics in the language pair English–Japanese in both directions. In both directions, the seven other BLEU scores are all statistically significantly different (p-value $<$ 0.05) from the BLEU score of the baseline system with a distortion limit of 6.

For the oracle experiments, all the scores are much higher than those of the baseline. The smallest improvement in extrinsic evaluation is in RIBES, around 6.5, when dl is equal to 6 in the language pair English to Japanese, but the difference is still statistically significant. The increase in BLEU scores is 4 points with a distortion limit of 0 and 3 points with a distortion limit of 6 in English to Japanese, 7 points with distortion limit of 0 and 5.5 points with distortion limit of 6 in Japanese to English, which is statistically significant. We also compare the results of the oracle experiments when the distortion limit is 0 to the baseline with a default distortion limit of 6. We get almost 2.5 BLEU point improvement in English to Japanese and 5 BLEU point improvement in Japanese to English. The oracle experiments outperform Nakagawa's top-down BTG parsing method, except in FRS and normalized τ scores for the language pair English to Japanese.

These results demonstrate the theoretical effectiveness of utilizing the tree structures generated by the HSSA method. In other words, the tree structures automatically generated using the HSSA method CAN benefit PB-SMT systems.

Our preordering model tries to reproduce the results of the oracle experiments. The scores for intrinsic evaluation metrics in both directions are better than those of the baseline, with large improvement. We obtain slight but statistically significant increases in the extrinsic evaluation with the same distortion limit. However, when compared to the baseline system with a default distortion limit of 6, the PB-SMT systems with a distortion limit of 0 that were built with our preordering models still lag behind, by around 1 BLEU point in English to Japanese and less than 0.5 BLEU point in Japanese

[5] http://www.logos.ic.i.u-tokyo.ac.jp/~tsuruoka/lapos/

	Language	Intrinsic		Intrinsic & Extrinsic		Extrinsic	
	pair	mod FRS	norm τ	RIBES		BLEU	
				dl = 0	dl = 6	dl = 0	dl = 6
Baseline	en-ja	51.12	73.99	65.83	68.10	19.45	21.51
Tree-based		66.12	83.08	69.31	70.11	20.43	21.97
Top-down		75.59	87.68	71.56	72.28	22.56	23.31
Oracle		66.60	87.39	75.17	74.74	23.75	24.23
Baseline	ja-en	59.41	72.98	64.87	65.87	16.01	18.10
Tree-based		64.87	80.14	66.23	66.63	17.55	18.76
Top-down		66.40	81.45	68.53	68.69	19.10	19.07
Oracle		68.18	85.81	75.44	75.18	23.20	23.87

Table 2: Intrinsic and extrinsic evaluation scores in English to Japanese and Japanese to English (mod FRS is the modified Fuzzy Reordering Score; norm τ is normalized Kendall's τ; dl stands for distortion limits). Baseline is a default PB-SMT system; Tree-based is our proposed preordering model; Top-down is the top-down BTG parsing-based reordering model; Oracle is an oracle system that uses HSSA tree structures obtained for the test set. The gray cells indicate the results to compare in translation: systems with preordering methods and with a distortion limit of 0 should be compared with the corresponding baseline system with a default distortion limit of 6; other results are given for completeness.

to English. However, the comparison is in favor of our system (preordering, distortion limit 0) in RIBES by 1 point. This seems natural as RIBES is a metric for machine translation which takes reordering into account.

The reasons for these mitigated results are listed below. Firstly, our preordering models do not simulates the HSSA method so well, because this method considers all words in the two parts at hand, while the learning models we used rely only on the features of two words in the beginning and the ending position of each part. Secondly, there may be several segmentation points with similar Ncut values when building the tree structures. We choose only one. To memorize other alternatives, the use of forests instead of trees would be required. Memorizing these alternatives may lead to larger increases in evaluation scores.

7 Conclusion

In this paper, we firstly automatically generate tree structures using the hierarchical sub-sentential alignment (HSSA) method. These tree structures are equivalent to parse trees obtained by Bracketing Transduction Grammars (BTG). Secondly, based on these tree structures, we build a preordering model. Thirdly, using this preordering model, source sentences are reordered. In an oracle experiment, we show that we may expect to outperform a baseline system with the default distortion limit of 6 by 2.5 (English to Japanese) or 5 (Japanese to English) BLEU points if we are able to reorder the text sentences exactly, without the need of any distortion limit. Other experiments show that tree structures generated by the HSSA method help in getting better RIBES scores than a baseline system without preordering.

In future work, we will try different features, times of iteration and sizes of beam. In addition, we would also like to try to the use of forest structures instead of tree structures.

Acknowledgements

The second author is supported in part by China Scholarship Council (CSC) under CSC Grant No. 201406890026. We would like to thank Tetsuji Nakagawa for his most helpful comments on the experiment setting details.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 529-536, Sydney, Australia, July. Association for Computational Linguistics.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4): 467-479.

Jingsheng Cai, Masao Utiyama, Eiichiro Sumita, and Yujie Zhang. 2014. Dependency-based Pre-ordering for Chinese-English Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 155-160, Baltimore, MD, USA, June. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2): 201-228.

Michael Collins and Brian Roark. 2004. Incremental Parsing with the Perceptron Algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 111-118, Barcelona, Spain, July. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause Restructuring for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531-540, Ann Arbor, MI, USA, June. Association for Computational Linguistics.

John DeNero and Jakob Uszkoreit. 2011. Inducing Sentence Structure from Parallel Corpora for Reordering. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 193-203, Edinburgh, Scotland, UK, July. Association for Computational Linguistics.

Nizar Habash. 2012. Syntactic Preprocessing for Statistical Machine Translation. In *Proceedings of the 11th Machine Translation Summit (MT-Summit)*, pages 215-222, Copenhagen, Denmark, September.

Dan Han, Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2012. Head Finalization Reordering for Chinese-to-Japanese Machine Translation. In *Proceedings of SSST-6, Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 57-66, Jeju, Korea, July. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 187-197, Edinburgh, Scotland, UK, July. Association for Computational Linguistics.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010a. Head Finalization: A Simple Reordering Rule for SOV Languages. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and Metrics MATR*, pages 244-251, Uppsala, Sweden, July. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010b. Automatic Evaluation of Translation Quality for Distant Language Pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944-952, MIT, Massachusetts, USA, October. Association for Computational Linguistics.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika* 30(1/2): 81-93.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language*, pages 48-54, Edmonton, Canada, May-June. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *2005 International Workshop on Spoken Language Translation*, pages 68-75, Pittsburgh, PA, USA, October.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177-180, Prague, Czech Republic, June. Association for Computational Linguistics.

Adrien Lardilleux, François Yvon, and Yves Lepage. 2012. Hierarchical Sub-sentential Alignment with Anymalign. In *Proceedings of the 16th annual conference of the European Association for Machine Translation (EAMT 2012)*, pages 279-286, Trento, Italy, May.

Uri Lerner and Slav Petrovs. 2013. Efficient Top-Down BTG Parsing for Machine Translation Preordering. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 513-523, Seattle, Washington, USA, October. Association for Computational Linguistics.

Percy Liang. 2005. Semi-supervised learning for natural language. Ph.D. Dissertation. Massachusetts Institute of Technology.

Tetsuji Nakagawa. 2015. Efficient Top-Down BTG Parsing for Machine Translation Preordering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 208-218, Beijing, China, July. Association for Computational Linguistics.

Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a Discriminative Parser to Optimize Machine Translation Reordering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 843-853, Jeju Island, Korea, July. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311-318, Philadelphia, PA, USA, July. Association for Computational Linguistics.

Hendra Setiawan, Min-Yen Kan and Haizhou Li. 2007. Ordering Phrases with Function Words. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 712-719, Prague, Czech Republic, June. Association for Computational Linguistics.

David Talbot, Hideto Kazawa, Hiroshi Ichikawa, Jason Katz-Brown, Masakazu Seno, and Franz J Och. 2011. A Lightweight Evaluation Framework for Machine Translation Reordering. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 12-21, Edinburgh, Scotland, UK, July. Association for Computational Linguistics.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of the 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (Short Papers)*, pages 101-104, Boston, MA, USA, May. Association for Computational Linguistics.

Roy Tromble and Jason Eisner. 2009. Learning Linear Ordering Problems for Better Translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007-1016, Singapore, August. Association for Computational Linguistics.

Yoshimasa Tsuruoka, Yusuke Miyao, and Junichi Kazama. 2011. Learning with Lookahead: Can History-Based Models Rival Globally Optimized Models?. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 238-246, Portland, Oregon, USA, June. Association for Computational Linguistics.

Karthik Visweswariah, Rajakrishnan Rajkumar, and Ankur Gandhe. 2011. A Word Reordering Model for Improved Machine Translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 486-496, Edinburgh, Scotland, UK, July. Association for Computational Linguistics.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 737-745, Prague, June. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3): 377-403.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Extracting Preordering Rules from Predicate-Argument Structures. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 29-37, Chiang Mai, Thailand, November.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 508-515, Geneva, Switzerland, August. Association for Computational Linguistics.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a Dependency Parser to Improve SMT for Subject-Object-Verb Languages. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pages 245-253, Boulder, Colorado, June. Association for Computational Linguistics.

Kenji Yamada and Kevin Knight. 2001. A Syntax-based Statistical Translation Model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523-530, Toulouse, France, July. Association for Computational Linguistics.

Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. 2001. Bipartite Graph Partitioning and Data Clustering. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 25-32, Atlanta, Georgia, USA, November. Association for Computational Linguistics.

Dongdong Zhang, Mu Li, Chi-Ho Li, and Ming Zhou. 2007. Phrase Reordering Model Integrating Syntactic Knowledge for SMT. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 533-540, Prague, Czech Republic, June. Association for Computational Linguistics.