Improving Fluency by Reordering Target Constituents Using MST Parser in English-to-Japanese Phrase-based SMT

Hwidong Na **Jin-Ji Li** **Jungi Kim** **Jong-Hyeok Lee**
Division of Electrical and Computer Engineering,
Pohang University of Science and Technology (POSTECH),
San 31 Hyoja Dong, Pohang, 790-784, Republic of Korea
{leona,ljj,yangpa,jhlee}@postech.ac.kr

Abstract

We propose a reordering method to improve the fluency of the output of the phrase-based SMT (PBSMT) system. We parse the translation results that follow the source language order into non-projective dependency trees, then reorder dependency trees to obtain fluent target sentences. Our method ensures that the translation results are grammatically correct and achieves major improvements over PBSMT using dependency-based metrics.

1 Introduction

Word order divergence is a major issue in phrase-based statistical machine translation (PBSMT). PBSMT assumes *symmetry* of structure in alignment heuristics. That is, the alignment heuristics (Och and Ney, 2004) used in PBSMT tend to grow diagonally from the intersection of bidirectional word alignments. If the word alignments are not symmetrical, the heuristics fail to find the correct word alignment. Also, the reordering model in PBSMT limits the movement of target phrases to a predefined window size. The heterogeneous structure between two languages requires long-distance movements which are impossible in PBSMT.

A pair of sentences in English and Japanese embeds mass structural divergences. For example, English follows an SVO (subject, verb, object) structure and uses prepositions for functional words, while Japanese follows an SOV structure and uses postpositions for functional words. As PBSMT is based on a distance-based word order model, the structural divergences between these languages lead to poor translation results. Moreover, the n-gram language model used in PBSMT ensures local fluency only, but not sentence-level fluency. Therefore, we need a global word order model to accommodate the structural divergences and enhance the fluency of the translation.

Previous research on global reordering focused on preprocessing and syntax-based approaches. Neither of the methods, however, employed word alignments directly for reordering. In this paper, we propose a novel method to reorder the target sentence as a postprocess of the PBSMT system. At the training step, we model the target language structure as per the source language. We then recover the original word order of the target language in the postprocessing stage.

2 Previous Work

Previous approaches to preprocessing have been focused on reordering the source sentence to follow the word order of the target sentence. Many researchers using PBSMT systems (with and without the distortion model after preprocessing) have tried to solve the global word order during preprocessing and have let the PBSMT adjust the local word order. Some have used manually built sets of rules to apply to the source parse tree (Collins et al., 2005; Li et al., 2009), and others have obtained the reordering statistics from training corpora using word alignment (Xia and McCord, 2004; Zhang et al., 2007). These methods are simple but effective especially for global reordering. However, they require an accurate and robust parser for the source language to minimize errors and to avoid parsing failure.

Syntax-based approaches are reliant on global reordering information embedded in the translation model using either the parse tree of source languages (Huang et al., 2006; Liu et al., 2006), target languages (Galley et al., 2004; Marcu et al., 2006; Liu and Gildea, 2008), or by using the parse trees of both source and target languages (Quirk et al., 2005). Syntax-based approaches integrated global reordering within an overall model. This process, however, increases the complexity of decoding and adds to the difficulty of error analysis.

Postprocessing approaches to the translation result have received little attention compared to the other approaches. Chang and Toutanova (2007) generated an n-best list using an n-gram language model with projective constraints for target languages in English to Japanese translation. The n-best list was reranked using a log-linear model with various syntactic features. They also modeled the global reordering model for target dependency trees with the local tree order model (LTOM). The LTOM assumes that orders of the local tree in the target dependency tree are independent, and that a dependent node has a relative offset to its head. Chang and Toutanova (2007) obtained a dependency tree of target languages by projecting the tree of source languages using heuristics as described in Quirk et al. (2005). The projective constraints enhance the fluency of the translation because the Japanese language almost always has projective dependency.

3 Proposed Method

At the training step, we create target sentences that follow the source language word order using the word alignment result. The word alignment encodes the structural divergence between the source and target languages and allows us to precisely reorder the target sentences. However, the word alignment process is only available at the training phase, but not at the decoding phase. Hence, previous preprocessing approaches cannot utilize word alignment to reorder the source sentences, and the preprocessing of a source sentence is typically undertaken without consideration of the corresponding target sentences. In our method, we reorder the target sentence according to the word alignment, and refer this operation as *symmetrize*.

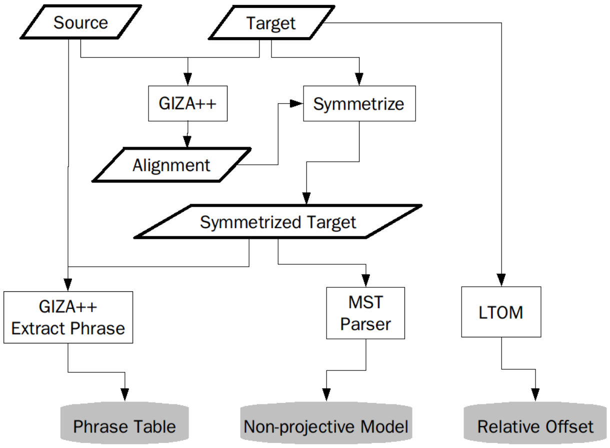


Figure 1: System architecture for training. Note the dependency trees of the target sentences are maintained even after symmetrization.

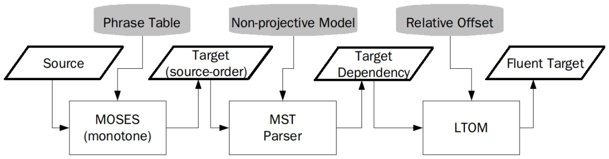


Figure 2: System architecture for decoding

Without the distortion model, the output of PB-SMT follows the word order of the source language. Therefore, a postprocessing process is required for global reordering in order to enhance the fluency of the output sentence. Our postprocessing method utilizes a dependency parsing. Because the dependency of the target sentence in source order would not be projective, we adopt a non-projective dependency parsing (McDonald et al., 2005). As the dependency trees are unordered, global reordering is induced by adjusting the parse trees. We also take advantage of projective constraints of the target language, as is done in Chang and Toutanova (2007). Unlike their method, however, our method directly parses the translation result.

Figures 1 and 2 show the overall architecture of our method for training and decoding. The symmetrized target sentences are used to train both PB-SMT and MST parser[1]. The original target sentences

[1] Maximum spanning tree (MST) parser finds the MST from a directed graph which is fully connected from one node to another.

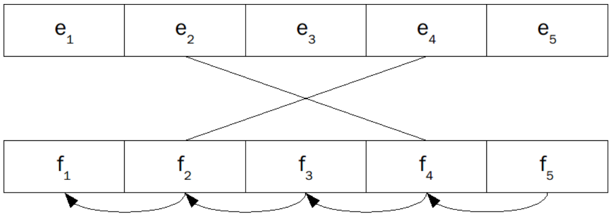
provides dependency order statistics used to train LTOM for adjusting unordered target sentences. As postprocessing, the result from PBSMT is parsed with the MST parser. The LTOM then adjusts the parse tree in order to enhance the fluency of the final translation.

3.1 Constituent in Japanese

A constituent in Japanese is a syntactic unit larger than a morpheme or a word, and is composed of content and functional words. Content words contain the (partial) meaning of a constituent. Functional words combine with content words to represent the whole meaning of a constituent. As a representative unit of meaning, it is more reasonable to build a dependency structure of constituents rather than words, especially in agglutinative languages such as Japanese or Turkish (Eryiğit et al., 2008).

In this paper, we denote a source language sentence as $E = e_1 \dots e_I$ and a target language sentence as $F = f_1 \dots f_J$, where I and J are the numbers of words in the source and target sentences, respectively. Content and functional words are grouped as constituents $\bar{F} = \bar{f}_1 \dots \bar{f}_K$, where K is the number of the constituents. A constituent $\bar{f}_k$ is composed of $f_{j_1} \dots f_{j_l}$, where $j_1 \dots j_l \in [1, J]$ and l is the number of words in a constituent $\bar{f}_k$. Then a dependency structure $H = h_1 \dots h_K$ is identified using a target language parser[2]. A head h_k is zero if $\bar{f}_k$ is the root and the head of $\bar{f}_k$ otherwise.

From the perspective of global word order, a constituent is a reordering unit in Japanese. A word alignment matrix $A = \{(i,j)|i \in [1, I], j \in [1, J]\}$ gives a constituent alignment $\bar{A} = \bar{a}_1 \dots \bar{a}_K$ by selecting one of the alignment of words for each constituent. As content words typically appear on the source side, we regard an alignment of content words as a constituent alignment. This enables us to obtain an accurate alignment. On the other hand, the word alignment of functional words have a low accuracy because functional words tend to mismatch. Let the words of the source sentence be $span(content(\bar{f}_k))$, where $content(\bar{f}_k)$ is the corresponding content words in a constituent of the target language. We assume that $span(content(\bar{f}_{k_1}))$ and $span(content(\bar{f}_{k_2}))$ do not overlap for all $k_1 \neq k_2$. Hence, any alignment from a constituent can be $\bar{a}_k$. In this paper, we select the smallest source index (min) among the word alignment, i.e., $\bar{a}_k = \min \{i|(i,j) \in A \cap j \in [j_1, j_l]\}$. $\bar{a}_k$ is zero if $\{(i,j)|j \in [j_1, j_l]\} = \emptyset$.



$$\begin{aligned} \bar{A}_{borrow} &= 04422 \\ \bar{A}_{inherit} &= 44220 \\ \bar{A}_{borrow-inherit} &= 44422 \\ \bar{A}_{inherit-borrow} &= 44222 \end{aligned}$$

Figure 3: Examples of constituent alignments $\bar{A}$ of borrow, inherit, borrow-inherit, and inherit-borrow methods

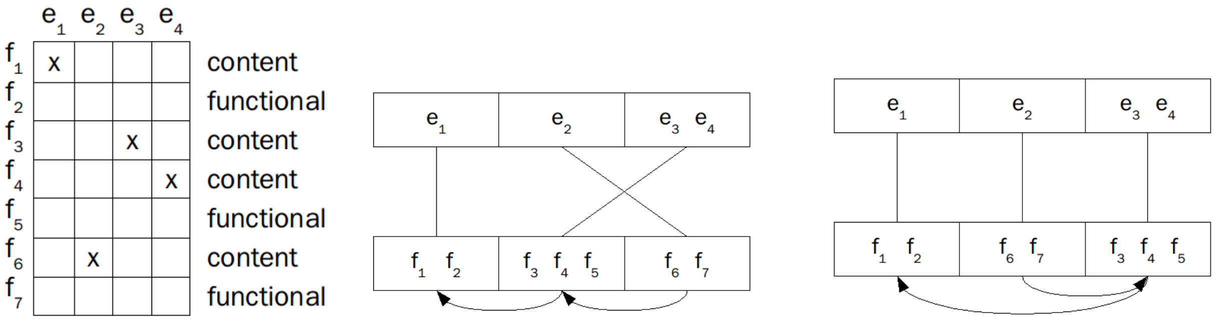
3.2 Training: Symmetrize by Reordering Target Constituents

For a constituent having no alignment ($\bar{a}_k = 0$), we examine two heuristics and their sequential combinations. Some word alignments need to be inserted due to word alignment error, especially in verb phrases. Examples are shown in Figure 3. Note that we have different results on combinations according to which method is applied first.

- borrow: Constituent borrows alignments from its children (bottom-up)
- inherit: Constituent inherits alignments from its parent (top-down)
- borrow-inherit: After borrowing, a constituent (leaf node) inherits alignments from its parent
- inherit-borrow: After inheriting, a constituent (root node) borrows alignments from its children

After selecting a constituent alignment, we sort target constituents by $\bar{a}_k$ to symmetrize the target sen-

[2] We use a Japanese dependency analyzer based on support vector machine (CaboCha) and regard parse trees as gold standard



(a) Word Alignment

(b) Original Corpus ($H = 230$)

(c) Symmetrized Corpus ($H' = 302$)

Figure 4: The word alignment (a) for three target constituents ($\bar{f}_1 = f_1 f_2$, $\bar{f}_2 = f_3 f_4 f_5$ and $\bar{f}_3 = f_6 f_7$) of a sample corpus (b) and the result of symmetrizing the target corpus (c). Note the non-projective dependency caused by symmetrization. Relative offsets $(\bar{f}_2, -1)$ and $(\bar{f}_1, -1)$ are learned for a head $\bar{f}_3$ and a head $\bar{f}_2$, respectively.

tence. Then symmetrized target constituents $\bar{F}'$ follows source language order. Note that we still maintain the target dependency tree of symmetrized target constituents, denoted H'.

For example, suppose we have a source sentence $E = e_1 \dots e_4$, a target sentence $F = f_1 \dots f_7$, and a word alignment $A = \{(1,1), (3,3), (4,4), (6,2)\}$ (Figure 4.(a)). By grouping $\bar{f}_1 = f_1 f_2$, $\bar{f}_2 = f_3 f_4 f_5$ and $\bar{f}_3 = f_6 f_7$, we have target constituents $\bar{F} = \bar{f}_1 \bar{f}_2 \bar{f}_3$, a constituent alignment $\bar{A} = 132$, and a constituent dependency $H = 230$. After symmetrizing as in Figure 4, we have $\bar{F}' = \bar{f}'_1 \bar{f}'_2 \bar{f}'_3 = \bar{f}_1 \bar{f}_3 \bar{f}_2$, and $H' = 302$.

3.3 Postprocessing: Non-projective Dependency Parsing using MST Parser

A symmetrized sentence in the target language corpus follows the word order of the source language. Unfortunately, we do not have a parser to process such an ungrammatical sentence. In addition, a symmetrized target sentence may contain non-projective dependency as shown in Figure 4.(c). MST parser for non-projective dependency takes advantage of $O(n^2)$ time complexity and robustness on parsing failure. Therefore, we adopt MST parser to train and parse symmetrized target corpus.

The annotated data for training MST parser is the symmetrized dependency H' of the original H as described in the previous section. We obtain the original dependency from a target language parser. For a given source sentence E, PBSMT without the distortion model translates E into a target sentence F'. Note that F' follows the source language order. Then, we group F' into target constituents $\bar{F}'$ using the target language analyzer and regard a constituent $\bar{f}'_k$ as a node of the directed graph. Finally a non-projective parsing gives an unordered dependency tree $\bar{H}'$.

3.4 Postprocessing: Adjust Dependency Tree

We obtain a fluent target language sentence by adjusting a non-projective unordered dependency tree given by the MST parser. Recall that Quirk et al. (2005) used a LTOM of the lexical rule and Chang and Toutanova (2007) used an additional n-gram language model to generate n-best order with projective constraints of the target language.

In this paper, we use a similar representation of LTOM, which chooses relative offsets of dependents to the head. From the target training corpus, LTOM learns relative offset statistics. In the previous example, relative offsets $(\bar{f}_2, -1)$ and $(\bar{f}_1, -1)$ are learned for a head $\bar{f}_3$ and a head $\bar{f}_2$, respectively.

In the test phase, LTOM chooses relative offsets of dependents for each head node of the MST. Reading off the dependency tree according to relative offsets, a target sentence eventually follows grammatical order of the target language. For instance, if a translation result $\bar{F}' = \bar{f}'_1 \bar{f}'_2 \bar{f}'_3 = \bar{f}_1 \bar{f}_3 \bar{f}_2$ has an unordered dependency $H' = 302$, then relative offsets of $\bar{f}_2$ for head $\bar{f}_3$, and $\bar{f}_1$ for head $\bar{f}_2$ are both -1. Reading off dependency tree gives $\bar{f}_1 \bar{f}_2 \bar{f}_3$.

Table 1: Corpus Usage and the number of sentences

	Train	Dev	Test
GIZA++	Constituent Alignment	Constituent Alignment	Not Available
PBSMT	Phrase Table	Parameter Tuning(MERT)	Monotone Decoding
MST Parser	Non-projective Model	Pseudo Parsing Accuracy	Dependency Parsing
LTOM	Relative Offset	Not Used	Tree Adjustment
# of sentences	1,172,709	609	1,381

We only use functional words in constituents (rather than whole words) to avoid data sparseness. For example, $(\bar{f}_2, -1) = (f_5, -1)$ and $(\bar{f}_1, -1) = (f_2, -1)$ are learned for a head $\bar{f}_3 = f_7$ and a head $\bar{f}_2 = f_5$, respectively. We also use the relative frequency of dependents given a head. When we do not have a certain trained offset for a head, we set the default offset of all dependents to -1 since Japanese is a head final language.

4 Experiments

4.1 System Description

In this paper, we use a patent translation corpus provided by NTCIR-7 Patent Translation Task[3]. The English corpus is lowercased and the Japanese corpus is segmented by morpheme. We convert wide alphanumerics in Japanese to half width. We use CaboCha[4] to parse the original target corpus.

The Baseline system uses an implementation of the PBSMT system, Moses[5], with the SRILM toolkit[6]. Both the Baseline and proposed method (Constituent) systems use trigram language model and minimum error rate training (MERT) included in the Moses toolkit.

First we obtain the bidirectional word alignments using GIZA++ and regard the intersection as the word alignment. Note that we only use the word alignment of content words of the target language and select the constituent alignment as described in Section 3. Then, we symmetrize the target corpus using the constituent alignment. A phrase table is trained using source and symmetrized target corpus. We use MERT with monotone decoding.

Table 2: Pseudo parsing unlabeled accuracy (UA) and unlabeled complete correct (UCC) for each alignment method (AM)

AM	UA	UCC
min	75.49	28.04
inherit	75.54	26.27
borrow	74.85	24.63
borrow-inherit	76.34	27.09
inherit-borrow	75.01	26.93
min*	93.06	58.62

Using the symmetrized target corpus, we use MST parser[7] for non-projective dependency. 10,000 sentences from the symmetrized target corpus is used to train the MST parser. LTOM is trained for relative offsets using the original target corpus with dependency.

During the test phase of the Constituent system, a source sentence is translated without the distortion model using Moses. After the translated sentences are grouped into constituents following source order, the MST parser creates an unordered dependency tree of the target sentence. The LTOM adjusts this parse tree by setting relative offsets of dependents for each head. Finally, a fluent target language sentence is achieved by reading off the parse tree. Corpus usage is summarized in Table 1.

4.2 Pseudo MST Parsing Accuracy

The pseudo dependency accuracy of the MST parser is estimated using a symmetrized development corpus. We call this "pseudo" because we regard the original parse tree from CaboCha as the gold standard. Although we have symmetrized the target constituents, parsing accuracy is measured to compare with the dependency tree of the symmetrized gold

[3] http://if-lab.slis.tsukuba.ac.jp/fujii/ntc7patmt/index-en.html

[4] http://www.chasen.org/~taku/software/cabocha/

[5] http://www.statmt.org/moses/

[6] http://www.speech.sri.com/projects/srilm/

[7] http://ryanmcd.googlepages.com/MST Parser.html

Table 3: Automatic evaluation results using n-gram based metrics, including the evaluation result using baseline without the distortion model at the decoding phase (Baseline w/o).

System	DevBLEU	BLEU	NIST
Baseline w/o	26.65	24.97	7.1102
Baseline		25.76	7.1676
Constituent	29.92	26.25	6.9414

standard.

We explore four types of alignment methods to improve the pseudo parsing accuracy, inherit and borrow methods, and their combinations as described in Section 3. Table 2 shows that the borrow-inherit method gives the best performance in terms of unlabeled accuracy. Hence, we use the borrow-inherit method for the Constituent method.

The oracle (min*) accuracy comes from the MST parser using the original target corpus. Instead of the symmetrized target corpus, the oracle MST trains the original training corpus and parses the original development corpus. The oracle accuracy (93%) is high enough to conclude that the training corpus does not suffer from data sparseness.

4.3 Automatic Evaluation

Human evaluators score the translation result with respect to adequacy and fluency, and regard the average of the two scores as translation quality. Callison-Burch et al. (2006) criticized n-gram based automatic evaluation metrics for weak correlation with human evaluation, especially with regards to fluency. Dependency-based automatic evaluation metrics have been developed to overcome the limitations of the n-gram based ones (Liu and Gildea, 2005; Owczarzak et al., 2007). They suggested metrics to evaluate machine translation results by parsing both translation results and reference sentences. Consequently, we use both n-gram based and dependency-based metrics in this paper.

Using n-gram based metrics, BLEU and NIST[8], we gain 0.49 BLEU but lose 0.23 NIST points (Table 3). On the development corpus, we gain 3.27 BLEU points. Using a dependency-based headword chain based metric (HWCM) proposed by Liu and Gildea (2005), we gain 3.94 and 4.54 points (Table 4). A headword chain is a sequence of nodes from a dependent to its ancestors in a dependency tree.

Table 4: Automatic evaluation results using a dependency based metric, using all word in a constituent and functional words only.

System	HWCM$_{all}$	HWCM$_f$
Baseline	11.22	29.80
Constituent	15.16	34.34

$$\text{HWCM} \quad = \quad \frac{1}{D}\sum_{d=1}^{D} \frac{\sum_{|c|=d} \#\text{ of } c \text{ in reference}}{\sum_{|c|=d} \#\text{ of } c},$$

where D is the maximum chain length and $|c|$ is the length of a chain c. By limiting the maximum chain length to 2, we have three unigram chains $\bar{f}_1$, $\bar{f}_2$, and $\bar{f}_3$ and three bigram chains $(\bar{f}_1, \bar{f}_2)$, $(\bar{f}_2, \bar{f}_3)$, and $(\bar{f}_3, root)$ in Figure 4.(b). We measure HWCM using all words (HWCM$_{all}$) and functional words only (HWCM$_f$) to represent a node of the dependency.

5 Discussion

5.1 Non-projective Dependency Parsing

Symmetrization leads to non-projective dependency. MST parser is trained to the symmetrized target corpus and gives a lower performance than the oracle parser as shown in Table 2. Despite the lower performance, we achieve a similar performance using the n-gram based automatic evaluation metric. Although the dependency-based metric shows significant improvements, our method still requires further refinement.

5.2 Tree Adjustment

The LTOM learns the dependency order statistics from the symmetrized target corpus at local level. To avoid data sparseness, we only use functional words from a constituent. Content words do not help tree adjustment, rather they encourage data sparseness. Experiments considering content words as units of relative offsets demonstrated lower scores than those using functional words only.

Table 5 shows an example of a translation result from the Baseline and from the proposed methods.

[8] We use mt-eval11b.pl. http://www.itl.nist.gov/iad/mig/tools/

Table 5: Translation examples. The main predicates (bold) require global reordering. In Japanese, brackets represent the boundaries of the constituents and functional words are underlined. The referenced sentence has five constituents and the dependency $H = 53550$, the result from the Baseline has four constituents and the dependency $H = 4340$, and the result from the Constituent has the dependency $H = 2440$.

System	Translation	BLEU	HWCM$_{all}$	HWCM$_f$
Source	an operation display section 42 **is provided** on the upper surface of the main body of the copying machine .			
Baseline	[操作 表示 部 42 が] **[設け られ て いる の]** [上面 に は 、] [複 機 本 で ある 。]	70.77	25.00	25.00
Constituent	[複 機 本 の] [上面 に は] [操作 表示 部 42 が] **[設け られ て いる 。]**	74.93	62.50	100.00
Reference	[また 、] [複 機 本 の] [上面 に は 、] [操作 表示 部 42 が] **[設け られ て いる 。]**			

The main predicate (**is provided**) on the source sentence needs global reordering to match a Japanese predicate (**設け られ て いる**) on the reference sentence. Our method outperforms the Baseline as the MST parser finds the predicate as the root and the LTOM accurately adjusts the unordered dependency. Note that the difference between two BLEU scores is small, though the translation result from the Baseline system is less grammatical than the result from the Constituent system. On the contrary, HWCM scores distinguish the fluency of the two results. Overall, HWCM scored higher fluency than BLEU.

5.3 N-gram versus dependency based automatic evaluation

Most automatic evaluation metrics treat functional words the same as content words. The target corpus we used contains an average of 16.74 content words, 9.9 functional words, and 8.71 constituents per sentence. A constituent contains an average of two content words. This leads to the high score on n-gram based automatic evaluation metrics. By decoding without the distortion model of the Baseline system and comparing the result with the distortion's, the difference is only 0.79 BLEU point and 0.0574 NIST score (Table 3).

Thus we need an orthogonal evaluation metric to avoid the pitfalls of an n-gram based one. Note that incorrect word order degrades translation quality even in a free-order language, such as Japanese. Global reordering therefore is clearly essential. The higher the parsing accuracy, the greater the fluency of a sentence. Consequently, we compare fluency of translation results using parsing accuracy. HWCM$_f$ shows the difference of fluency more clearly than HWCM$_{all}$. Because we use constituent-level dependency, identical constituents have slightly different content words. Table 5 shows that a constituent of our method (上面 に は) and one of reference (上面 に は 、) which are identical. Thus, HWCM$_f$ reflects fluency better than HWCM$_{all}$. On both HWCM metrics, Table 4 shows that our Constituent method is more fluent than the Baseline method.

6 Conclusion

We propose a novel method to improve the fluency of translation results. We symmetrized a target corpus using the word alignment at the training step. For global reordering of translation results, we parsed the result and adjusted the dependency tree during postprocessing. Despite the accuracy (potential) loss in each intermediate step, our method achieved a 1.90% relative improvement on BLEU scores compared to the Baseline system. We also gained 35.12% and 15.23% relative improvements on HWCM$_{all}$ and HWCM$_f$, respectively. We demonstrate that higher grammatical accuracy in the translation can be achieved by preserving the projective constraints in the Japanese language. We have much room to improve the proposed method. In the futures we wish to investigate the n-best reranking approaches within our framework.

Acknowledgments

We appreciate to anonymous reviewers for inspiring comments. This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MEST No. 2009-0075211), in part by the Microsoft Research Asia (MSRA), and in part by the BK 21 Project in 2009.

References

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of bleu in machine translation research. In *In EACL*, pages 249–256.

Pi-Chuan Chang and Kristina Toutanova. 2007. A discriminative syntactic word order model for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 9–16, Prague, Czech Republic, June. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 531–540, Morristown, NJ, USA. Association for Computational Linguistics.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of turkish. *Comput. Linguist.*, 34(3):357–389.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL-04)*, Boston, USA, May.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. A syntax-directed translator with extended domain of locality. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 1–8, New York City, New York, June. Association for Computational Linguistics.

Jin-Ji Li, Jungi Kim, Dong-Il Kim, and Jong-Hyeok Lee. 2009. Chinese syntactic reordering for adequate generation of Korean verbal phrases in Chinese-to-Korean SMT. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 190–196, Athens, Greece, March. Association for Computational Linguistics.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Ding Liu and Daniel Gildea. 2008. Improved tree-to-string transducer for machine translation. In *Proceedings of WMT*.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 609–616, Morristown, NJ, USA. Association for Computational Linguistics.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. Spmt: Statistical machine translation with syntactified target language phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 44–52, Sydney, Australia, July. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449.

Karolina Owczarzak, Josef Genabith, and Andy Way. 2007. Evaluating machine translation with lfg dependencies. *Machine Translation*, 21(2):95–119.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: syntactically informed phrasal smt. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 271–279, Morristown, NJ, USA. Association for Computational Linguistics.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 508, Morristown, NJ, USA. Association for Computational Linguistics.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1–8, Rochester, New York, April. Association for Computational Linguistics.