Constituent Reordering and Syntax Models for English-to-Japanese Statistical Machine Translation

Young-Suk Lee
IBM Research
ysuklee@us.ibm.com

Bing Zhao
IBM Research
zhaob@us.ibm.com

Xiaoqiang Luo
IBM Research
xiaoluo@us.ibm.com

Abstract

We present a constituent parsing-based reordering technique that improves the performance of the state-of-the-art English-to-Japanese phrase translation system that includes distortion models by 4.76 BLEU points. The phrase translation model with reordering applied at the pre-processing stage outperforms a syntax-based translation system that incorporates a phrase translation model, a hierarchical phrase-based translation model and a tree-to-string grammar. We also show that combining constituent reordering and the syntax model improves the translation quality by additional 0.84 BLEU points.

1 Introduction

Since the seminal work by (Wu, 1997) and (Yamada and Knight, 2001), there have been great advances in syntax-based statistical machine translation to accurately model the word order distortion between the source and the target languages.

Compared with the IBM source-channel models (Brown et al., 1994) and the phrase translation models (Koehn et al., 2003), (Och and Ney, 2004) which are good at capturing local reordering within empirical phrases, syntax-based models have been effective in capturing the long-range reordering between language pairs with very different word orders like Japanese-English (Yamada and Knight, 2001), Chinese-English (Chiang, 2005) and Urdu-English (Zollmann et al. 2008), (Callison-Burch et al. 2010).

However, (Xu et al., 2009) show that applying dependency parsing-based reordering as pre-processing (pre-ordering hereafter) to phrase translation models produces translation qualities significantly better than a hierarchical phrase-based translation model (Hiero hereafter) implemented in (Zollman and Venugopal, 2006) for English-to-Japanese translation. They also report that the two models result in comparable translation qualities for English-to-Korean/Hindi/Turkish/Urdu, underpinning the limitations of syntax-based models for handling long-range reordering exhibited by the strictly head-final Subject-Object-Verb (SOV) order languages like Japanese and the largely head-initial Subject-Verb-Object (SVO) order languages like English.

In this paper, we present a novel constituent parsing-based reordering technique that uses manually written context free (CFG hereafter) and context sensitive grammar (CSG hereafter) rules. The technique improves the performance of the state-of-the-art English-to-Japanese phrase translation system that includes distortion models by 4.76 BLEU points. The phrase translation model with constituent pre-ordering consistently outperforms a syntax-based translation system that integrates features from a phrase translation model, Hiero and a tree-to-string grammar. We also achieve an additional 0.84 BLEU point improvement by applying an extended set of reordering rules that incorporate new rules learned from the syntax model for decoding.

The rest of the paper is organized as follows. In Section 2, we summarize previous work related to this paper. In Section 3, we give an overview of the syntax model with which we compare the performance of a phrase translation

626

model with pre-ordering. We also discuss a chart-based decoder used in all of our experiments. In Section 4, we describe the constituent parsing-based reordering rules. We show the impact of pre-ordering on a phrase translation model and compare its performance with the syntax model. In Section 5, we discuss experimental results from the combination of syntax model and pre-ordering. Finally in Section 6, we discuss future work.

2 Related Work

Along the traditions of unsupervised learning by (Wu, 1997), (Chiang, 2005) presents a model that uses hierarchical phrases, Hiero. The model is a synchronous context free grammar learned from a parallel corpus without any linguistic annotations and is applied to Chinese-to-English translation. (Galley and Manning, 2008) propose a hierarchical phrase reordering model that uses shift-reduce parsing.

In line with the syntax-based model of (Yamada and Knight, 2001) that transforms a source language parse tree into a target language string for Japanese-English translation, linguistically motivated syntactic features have been directly incorporated into both modeling and decoding. (Liu, et. al. 2006), (Zhao and Al-Onaizan, 2008) propose a source tree to target string grammar (tree-to-string grammar hereafter) in order to utilize the source language parsing information for translation. (Liu, et. al. 2007) propose packed forest to allow ambiguities in the source structure for the tree-to-string grammar. (Ding and Palmer, 2005) and (Zhang et. al., 2006) propose a tree-to-tree grammar, which generates the target tree structure from the high-precision source syntax. (Shen, et. al., 2008) propose a string to dependency tree grammar to use the target syntax when the target is English for which parsing is more accurate than other languages. (Marcu et al., 2006) introduce a syntax model that uses syntactified target language phrases. (Chang and Toutanova, 2007) propose a global discriminative statistical word order model that combines syntactic and surface movement information, which improves the translation quality by 2.4 BLEU points in English-to-Japanese translation. (Zollmann, et. al., 2008) compare various translation models and report that the syntax augmented model works

better for Chinese-to-English and Urdu-to-English, but not for Arabic-to-English translation. (Carreras and Collins, 2009) propose a highly flexible reordering operations during tree adjoining grammar parsing for German-English translation. (Callison-Burch et al., 2010) report a dramatic impact of syntactic translation models on Urdu-to-English translation.

Besides the approaches which integrate the syntactic features into translation models, there are approaches showing improvements via pre-ordering for model training and decoding. (Xia and McCord, 2004), (Collins et al., 2005) and (Wang, et. al. 2007) apply pre-ordering to the training data according to language-pair specific reordering rules to improve the translation qualities of French-English, German-English and Chinese-English, respectively. (Habash, 2007) uses syntactic preprocessing for Arabic-to-English translation. (Xu et al., 2009) use a dependency parsing-based pre-ordering to improve translation qualities of English to five SOV languages including Japanese.

The current work is related to (Xu et al., 2009) in terms of the language pair and translation models explored. However, we use constituent parsing with hierarchical rules, while (Xu et al., 2009) use dependency parsing with precedence rules. The two approaches have different rule coverage and result in different word orders especially for phrases headed by verbs and prepositions. We also present techniques for combining the syntax model with tree-to-string grammar and pre-ordering for additional performance improvement. The total improvement by the current techniques over the state-of-the-art phrase translation model is 5.6 BLEU points, which is an improvement gap not attested elsewhere with reordering approaches.
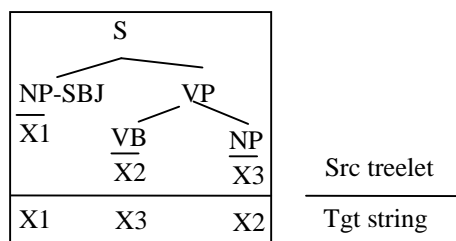
3 Syntax Model and Chart-Based Decoder

In this section, we give an overview of the syntax model incorporating a tree-to-string grammar. We will compare the syntax model performance with a phrase translation model that uses the pre-ordering technique we propose in Section 4. We also describe the chart-based decoder that we use in all of the experiments reported in this paper.

3.1 Tree-to-String Grammar

Tree-to-string grammar utilizes the source language parse to model reordering probabilities from a source tree to the target string (Liu et. al., 2006), (Liu et. al., 2007), (Zhao and Al-Onaizan, 2008) so that long distance word reordering becomes local in the parse tree.

Reordering patterns of the source language syntax and their probabilities are automatically learned from the word-aligned source-parsed parallel data and incorporated as a tree-to-string grammar for decoding. Source side parsing and the resulting reordering patterns bound the search space. Parsing also assigns linguistic labels to the chunk, e.g. NP-SBJ, and allows statistics to be clustered reasonably. Each synchronous context free grammar (SCFG) rewriting rule rewrites a source treelet into a target string, with both sides containing hiero-style variables. For instance, the rule [X, VP] [X, VB] [X,NP] → [X, NP] [X, VB] rewrites a VP with two constituents VB and NP into an NP VB order in the target, shown below.

```
                S
        ┌───────┴───────┐
     NP-SBJ            VP
       ──          ┌────┴────┐
       X1         VB        NP
                  ──        ──
                  X2        X3      Src treelet
    ─────────────────────────────  ─────────
       X1        X3        X2       Tgt string
```

The tree-to-string grammar introduces possible search space to generate an accurate word order, which is refined on the basis of supports from other models. However, if the correct word order cannot be generated by the tree-to-string grammar, the system can resort to rules from Hiero or a phrase translation model for extended rule coverage.

3.2 Chart-based Decoder

We use a chart-based decoder – a template decoder that generalizes over various decoding schemes in terms of the dot-product in Earley-style parsing (Earley, 1970) – to support various decoding schemes such as phrase, Hiero (Chiang, 2005), Tree-to-String, and the mixture of all of the above.

This framework allows one to strictly compare different decoding schemes using the same feature and parameter setups. For the experimental results in Sections 4 & 5, we applied (1) phrase decoding for the baseline phrase translation system that includes distortion models, (2) Hiero decoding for the Hiero system that incorporates a phrase translation model, and (3) Tree-to-string decoding for the syntax-based systems that incorporate features from phrase translation, Hiero and tree-to-string grammar models.

The decoder seeks the best hypothesis e^* according to the Bayesian decision rule (1):

$$e* = \operatorname*{argmin}_{\{e, d \in D\}} \phi(e) \cdot \phi(d) \qquad (1)$$

d is one derivation path, rewriting the source tree into the target string via the probabilistic synchronous context free tree-to-string grammar (PSCFG). $\phi(e)$ is the cost functions computed from general n-gram language models. In this work, we use two sets of interpolated 5-gram language models. $\phi(d)$ is a vector of cost functions defined on the derivation sequence. We have integrated 18 cost functions ranging from the basic relative frequencies and IBM model-1 scores to counters for different types of rules including blocks, glue, Hiero, and tree-to-string grammar rules. Additional cost functions are also integrated into the decoder, including measuring the function/content-word mismatch between source and target, similar to (Chiang et. al., 2009) and length distribution for non-terminals in (Shen et. al., 2009).

4 Parsing and Reordering Rules

We apply a set of manually acquired reordering rules to the parsing output from a constituent parser to pre-order the data for model training and decoding.

4.1 Parsing with Functional Tags

We use a maximum entropy English parser (Ratnaparkhi, 1999) trained on OntoNotes (Hovy, 2006) data. OntoNotes data include most of the Wall Street Journal data in Penn Treebank (Marcus et al., 1993) and additional data from broadcast conversation, broadcast news and web log.
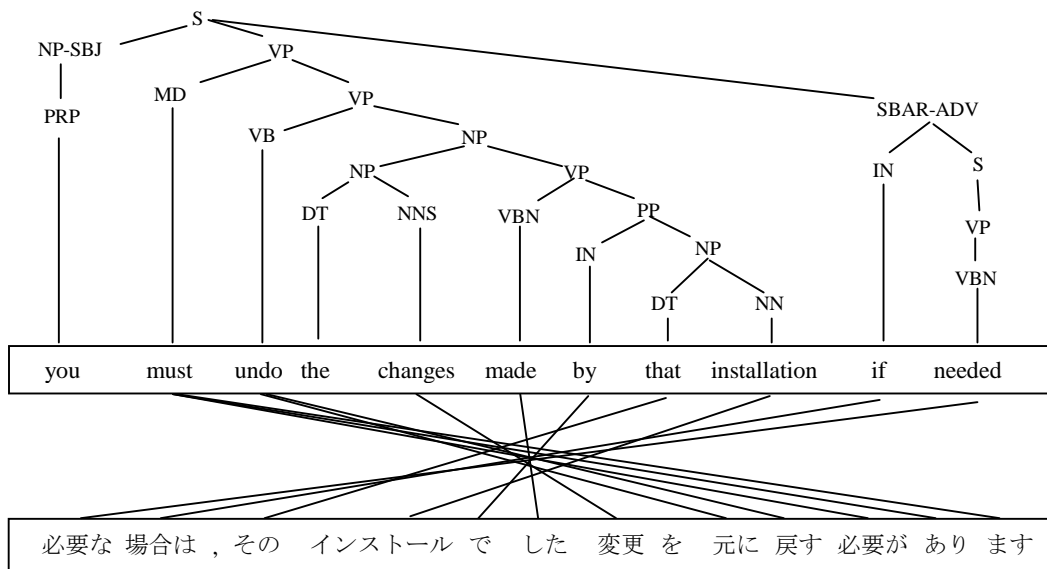
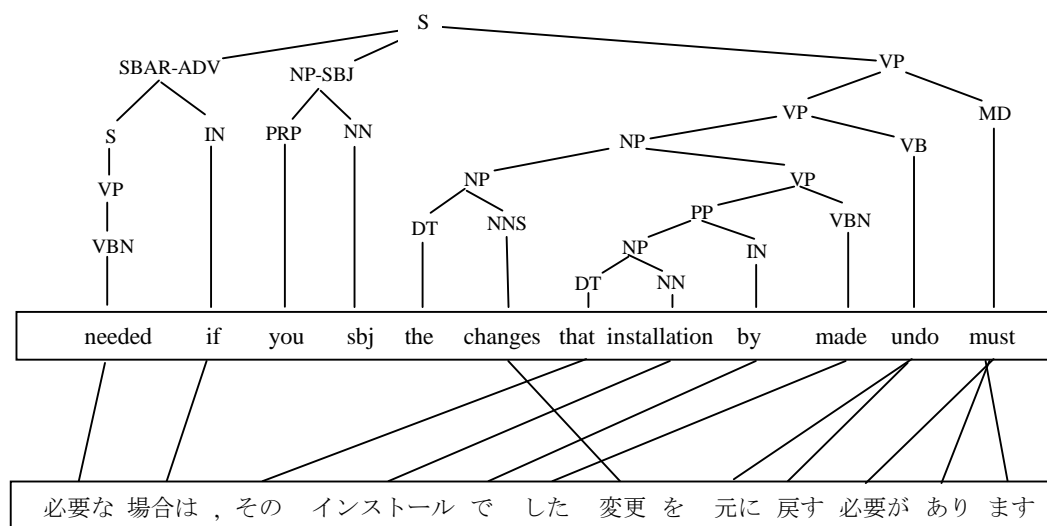Figure 1. Parse Tree and Word Alignment before Reordering

Figure 2. Parse Tree and Word Alignment after Reordering

The parser is trained with all of the functional and part-of-speech (POS) tags in the original distribution: total 59 POS tags and 364 phrase labels.

We use functional tags since reordering decisions for machine translation are highly influenced by the function of a phrase, as will be shown later in this section. An example parse tree with functional tags is given at the top half of Figure 1. NP-SBJ indicates a subject noun phrase, SBAR-ADV, an adverbial clause.

4.2 Structural Divergence between English and Japanese

Japanese is a strictly head-final language, i.e. the head is located at the end of a phrase. This leads to a high degree of distortions with English, which is largely head initial.

The word order contrast between the two languages is illustrated by the human word alignment at the bottom half of Figure 1. All instances of word alignments are non-monotonic except for the sequence *that installation*, which is monotonically aligned to the Japanese morpheme sequence その インストール. Note that there are no word boundaries in Japanese written text, and we apply Japanese morpheme segmentation to obtain morpheme sequences in the figure. All of the Japanese examples in this paper are presented with morpheme segmentation.

The manual reordering rules are written by a person who is proficient with English and Japanese/Korean grammars, mostly on the basis of perusing parsed English texts.

4.3 CFG Reordering Rules

Our reordering rules are mostly CFG rules and divided into head and modifier rules.

Head reordering rules in Table 1 move verbs and prepositions from the phrase initial to the phrase final positions (Rules 1-11). Reordering of the head phrase in an adverbial clause also belongs to this group (Rules 12-14). The label sequences in **Before RO** and **After RO** are the immediate children of the **Parent Node** before and after reordering. VBX stands for VB, VBZ, VBP, VBD, VBN and VBG. XP^+ stands for one or more POS and/or phrase labels such as MD, VBX, NP, PP, VP, etc. In 2 & 4, RB is the tag for negation *not*. In 5, RP is the tag for a verb particle.

Modifier reordering rules in Table 2 move modified phrases from the phrase initial to the phrase final positions within an NP (Rules 1-3). They also include placement of NP, PP, ADVP within a VP (Rules 4 & 5). The subscripts in a rule, e.g. PP_1 and PP_2 in Rule 3, indicate the distinctness of each phrase sharing the same label.

4.4 CSG Reordering Rules

Some reordering rules cannot be captured by CFG rules, and we resort to CSG rules.[1]

	Parent Node	Before RO	After RO
1	VP	MD VP	VP MD
2	VP	MD RB VP	VP MD RB
3	VP	VBX XP^+	XP^+ VBX
4	VP	VBX RB XP^+	XP^+ VBX RB
5	VP	VBX RP XP^+	XP^+ VBX RP
6	ADJP-PRD	JJ XP^+	XP^+ JJ
7	PP	IN NP	NP IN
8	PP	IN S	S IN
9	SBAR-TMP	IN S	S IN
10	SBAR-ADV	IN S	S IN
11	SBAR-PRP	IN S	S IN
12	SBAR-TMP	WHADVP S	S WHADVP
13	SBAR-ADV	WHADVP S	S WHADVP
14	SBAR-PRP	WHADVP S	S WHADVP

Table 1. Head Reordering Rules

	Parent Node	Before RO	After RO
1	NP	NP SBAR	SBAR NP
2	NP	NP PP	PP NP
3	NP	NP PP_1 PP_2	PP_1 PP_2 NP
4	VP	VBX NP PP	PP NP VBX
5	VP	VBX NP ADVP-TMP PP	PP NP ADVP-TMP VBX

Table 2. Modifier Reordering Rules

For instance, in the parse tree and word alignment in Figure 1, the last two English words *if needed* under SBAR-ADV is aligned to the first two Japanese words 必要な 場合は. In order to change the English order to the corresponding Japanese order, SBAR-ADV dominated by the VP should move across the VP to sentence initial position, as shown in the top half of Figure 2, requiring a CSG rule.

The adverbial clause reordering in Figure 2 is denoted as Rule 1 in Table 3, which lists two other CSG rules, Rule 2 & 3.[2] The subscripts in Table 3 are interpreted in the same way as those in Table 2.

---

[1] These CSG rules apply to trees of depth two or more, and the applications are dependent on surrounding contexts. Therefore, they are different from CFG rules which apply only to trees of depth one, and TSG (tree substitution grammar) rules for which variables are independently substituted by substitution. The readers are referred to

(Joshi and Schabes, 1997) for formal definitions of various grammar formalisms.

[2] Rule 3 is applied after all CFG rules, see Section 4.6. Therefore, VBX's are located at the end of each corresponding VP.

	Before RO After RO
1	(S XP$_1^+$ (VP XP$_2^+$ SBAR-ADV))
	(S SBAR-ADV XP$_1^+$ (VP XP$_2^+$))
2	(S XP$_1^+$ (VP (XP$_2^+$ SBAR-ADV)))
	(S XP$_1^+$ SBAR-ADV (VP (XP$_2^+$)))
3	(VP$_1$ ADVP-MNR (VP$_2$ XP$^+$ VBX$_2$) VBX$_1$)
	(VP$_1$ (VP$_2$ XP$^+$ ADVP-MNR VBX$_2$) VBX$_1$)

Table 3. CSG Reordering Rules

ADVP-MNR stands for a manner adverbial phrase such as *explicitly* in the following: *The software version has been explicitly verified as working*. Rule 3 in Table 3 indicates that a ADVP-MNR has to immediately precede a verb in Japanese, resulting in the substring '...*as working explicitly verified...*' after reordering.

Note that functional tags allow us to write reordering rules specific to semantic phrases. For instance, in Rule 1, SBAR-ADV under VP moves to the sentence initial position under S, but an SBAR without any functional tags do not. It typically stays within a VP as the complement of the verb.

4.5 Subject Marker Insertion

Japanese extensively uses case particles that denote the role of the preceding noun phrase, for example, as subject, object, etc. We insert *sbj*, denoting the subject marker, at the end of a subject noun phrase NP-SBJ. The inserted subject marker *sbj* mostly gets translated into the subject particle が or は in Japanese.[3]

4.6 Reordering Rule Application

The rules are applied categorically, sequentially and recursively. CSG Rules 1 and 2 in Table 3 are applied before all of the CFG rules. Among CFG rules, the modifier rules in Table 2 are applied before the head rules in Table 1. CSG Rule 3 in Table 3 is applied last, followed by the subject marker insertion operation.

CFG head and modifier rules are applied recursively. The top half of Figure 2 is the parse tree obtained by applying reordering rules to the parse tree in Figure 1. After reordering, the word alignment becomes mostly monotonic, as shown at the bottom half of Figure 2.

---

[3] The subject marker insertion is analogous to the insertion operation in (Yamada and Knight, 2001), which covers a wide range of insertion of case particles and verb inflections in general.

4.7 Experimental Results

All systems are trained on a parallel corpus, primarily from the Information Technology (IT) domain and evaluated on the data from the same domain. The training data statistics is in Table 4 and the evaluation data statistics is in Table 5. Japanese tokens are morphemes and English tokens are punctuation tokenized words.

Corpus Stats	English	Japanese
sentence count	3,358,635	3,358,635
token count	57,231,649	68,725,865
vocabulary size	242,712	348,221

Table 4. Training Corpus Statistics

Data Sets	Sentence Count	Token Count
Tuning	600	11,761
DevTest	437	8,158
Eval	600	11,463

Table 5. Evaluation Data Statistics

We measure the translation quality with IBM BLEU (Papineni et al., 2002) up to 4 grams, using 2 reference translations, BLEUr2n4. For BLEU score computation, we character-segment Kanji and Kana sequences in the reference and the machine translation output. Various system performances are shown in Table 6.

Models	Tuning	DevTest	Eval
Phrase (BL)	0.5102	0.5330	0.5486
Hiero	0.5385	0.5574	0.5724
Syntax	0.5561	0.5777	0.5863
Phrase+RO1	0.5681	0.5793	0.5962

Table 6. Model Performances (BLEUr2n4)

Phrase (BL) is the baseline phrase translation system that incorporates lexical distortion models (Al-Onaizan and Papineni, 2006). Hiero is the hierarchical phrase-based system (Chiang, 2006) that incorporates the phrase translation model. Syntax is the syntax model described in Section 3, which incorporates the phrase translation, Hiero and tree-to-string grammar models. Phrase+RO1 is the phrase translation model with pre-ordering for system training and decoding, using the rules described in this section. Phrase+RO1 improves the translation quality of the baseline model by 4.76 BLEU points and outperforms the syntax model by over 0.9 BLEU points.

5 Constituent Reordering and Syntax Model Combined

Translation qualities of systems that combine the syntax model and pre-ordering are shown in Table 7. Syntax+RO1 indicates the syntax model with pre-ordering discussed in Section 4. Syntax+RO2 indicates the syntax model with a more extensive pre-ordering for decoding discussed below .

Models	Tuning	DevTest	Eval
Phrase+RO1	0.5681	0.5793	0.5962
Syntax+RO1	0.5742	0.5802	0.6003
Syntax+RO2	0.5769	0.5880	0.6046

Table 7. Syntax Model with Pre-ordering

Analyses of the syntax model in Table 6 revealed that automatically learned rules by the tree-to-string grammar include new rules not covered by the manually written rules, some of which are shown in Table 8.

Parent Node	Before RO	After RO
ADJP-PRD	RB JJ PP	PP RB JJ
ADVP-TMP	RB PP	PP RB
ADVP	ADVP PP	PP ADVP
NP	NP VP	VP NP

Table 8. New CFG rules automatically learned by Tree-to-String grammar

We augment the manual rules with the new automatically learned rules. We call this extended set of reordering rules RO2. We use the manual reordering rules RO1 for model training, but use the extended rules RO2 for decoding. And we obtain the translation output Syntax+RO2 in Table 7. Syntax+RO2 outperforms Phrase+RO1 by 0.84 BLEU points, and Syntax+RO1 by 0.43 BLEU points.

In Table 9, we show the ratio of each rule type preserved in the derivation of one-best translation output of the following two models: Syntax   and Syntax+RO2.   In the table, 'Blocks' indicate phrases from the phrase translation model and 'Glue Rules' denote the default grammar rule for monotone decoding.

The syntax model without pre-ordering (Syntax) heavily utilizes the Hiero and tree-to-string grammar rules, whereas the syntax model with pre-ordering (Syntax+RO2) mostly depends on monotone decoding with blocks and glue rules.

Rule Type	Syntax	Syntax+RO2
Blocks	46.3%	51.2%
Glue Rules	6.0%	37.3%
Hiero Rules	18.3%	1.3%
Tree-to-String	29.4%	10.2%

Table 9. Ratio of each rule type preserved in the translation derivation of Syntax and Syntax+RO2

6 Summary and Future Research

We have proposed a constituent pre-ordering technique for English-to-Japanese translation. The technique improves the performance of the state-of-the-art phrase translation models by 4.76 BLEU points and outperforms a syntax-based translation system that incorporates a phrase translation model, Hiero and a tree-to-string grammar. We have also shown that combining constituent pre-ordering and the syntax model improves the translation quality by additional 0.84 BLEU points.

While achieving solid performance improvement over the existing translation models for English-to-Japanese translation, our work has revealed some limitations of syntax models both in terms of grammar representations and modeling.   Whereas many syntax models are based on CFG rules for probability acquisition, the current research shows that there are various types of reordering that require the generative capacity beyond CFG. While most of the reordering rules for changing the English order to the Japanese order (and vice versa) should apply categorically,[4] often the probabilities of tree-to-string grammar rules are not high enough to survive in the translation derivations.

As for the reordering rules that require the generative capacity beyond CFG, we may model mildly context sensitive grammars such as tree adjoining grammars (Joshi and Schabes, 1997), as in (Carreras and Collins, 2009). The

---

[4] Assuming that the parses are correct, the head reordering rules in Table 1 have to apply categorically to change the English order into the Japanese order because English is head initial and Japanese is head final without any exceptions. Similarly, most of the modifier reordering rules in Table 2 have to apply categorically because most modifiers follow the modified head phrase in English, e.g. a relative clause modifier follows the head noun phrase, a prepositional phrase modifier follows the head noun phrase, etc., whereas modifier phrases precede the modified head phrases in Japanese.

extended domain of locality of tree adjoining grammars should suffice to capture non-CFG reordering rules for many language pairs. Alternatively, we can adopt enriched feature representations so that a tree of depth one can actually convey information on a tree of several depths, such as parent annotation of (Klein and Manning, 2003).

Regarding the issue of modeling, we can introduce a rich set of features, as in (Ittycheriah and Roukos, 2007), the weights of which are trained to ensure that the tree-to-string grammar rules generating the accurate target orders are assigned probabilities high enough not to get pruned out in the translation derivation.

Acknowledgement

We would like to acknowledge IBM RTTS (Realtime Translation Systems) team for technical discussions on the topic and the provision of linguistic resources. We also would like to thank IBM SMT (Statistical Machine Translation) team for various software tools and the anonymous reviewers for their helpful comments .

References

Y. Al-Onaizan and K. Papineni. 2006. Distortion models for statistical machine translation. *Proceedings of ACL-COLING*. Pages 529-536.

C. Baker, S. Bethard, M. Bloodgood, R. Brown, C. Callison-Burch, G. Coppersmith, B. Dorr, W. Filardo, K. Giles, A. Irvine, M. Kayser, L. Levin, J. Martineau, J. Mayfield, S. Miller, A. Phillips, A. Philpot, C. Piatko, L. Schwartz, D. Zajic. 2010. Semantically Informed Machine Translation (SIMT). Final Report of the 2009 Summer Camp for Applied Language Exploration.

P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation, *Computational Linguistics*, 19(2):263–311.

X. Carreras and M. Collins. 2009. Non-projective parsing for statistical machine translation. *Proceedings of the 2009 EMNLP*. Pages 200-209.

P. Chang and C. Toutanova. 2007. A Discriminative Syntactic Word Order Model for Machine Translation. *Proceedings of ACL*. Pages 9-16.

D. Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. *Proceedings of ACL*. Pages 263-270.

D. Chiang, W. Wang and Kevin Knight. 2009. 11,001 new features for statistical machine translation. *Proceedings of HLT-NAACL*. Pages 218-226.

M. Collins, P. Koehn, I. Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. *Proceedings of ACL*. Pages 531-540.

Y. Ding and M. Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. *Proceedings of ACL*. Pages 541-548.

J. Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*. Vol. 13. Pages 94–102.

M. Galley and C. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. *Proceedings of EMNLP*.

N. Habash. 2007. Syntactic Preprocessing for Statistical Machine Translation. *Proceedings of the Machine Translation Summit*.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw and R. Weischedel. 2006. OntoNotes: The 90% Solution. *Proceedings of HLT*. Pages 57-60.

A. Ittycheriah and S. Roukos. 2007. Direct Translation Model 2. *Proceedings of HLT-NAACL*.

A. Joshi and Y. Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and K. Salomaa, editors, *Handbook of Formal Languages*, volume 3. Springer.

D. Klein and C. Manning. 2003. Accurate Unlexicalized Parsing. *Proceedings of 41st ACL*.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation, *Proceedings of HLT−NAACL*. Pages 48−54.

Y. Liu, Q. Liu and S. Lin. 2006. Tree-to-string alignment template for statistical machine translation. *Proceedings of ACL-COLING*.

Y. Liu, Y. Huang, Q. Liu, and S. Lin. 2007. Forest-to-string statistical translation rules. *Proceedings of the 45th ACL*.

D. Marcu, W. Wang, A. Echihabi and K. Knight. 2006. SPMT: Statistical Machine Translation with Syntactified Target Language Phrases. *Proceedings of EMNLP*. Pages 44-52.

M. Marcus, B. Santorini and M. Marcinkiewicz. 1993. Building a Large Annotated Corpus of Eng-

lish: the Penn Treebank. *Computational Linguistics*, 19(2): 313-330.

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*: Vol. 30. Pages 417– 449.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Proceddings of ACL.* Pages 311–318.

A. Ratnaparkhi. 1999. Learning to Parse Natural Language with Maximum Entropy Models. *Machine Learning: Vol. 34*. Pages 151-178.

L. Shen, J. Xu and R. Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. *Proceedings of ACL.*

L. Shen, J. Xu, B. Zhang, S. Matsoukas and Ralph Weischedel. 2009. Effective Use of Linguistic and Contextual Information for Statistical Machine Translation. *Proceedings of EMNLP.*

C. Wang, M. Collins, P. Koehn. 2007. Chinese Syntactic Reordering for Statistical Machine Translation. *Proceedings of EMNLP-CoNLL.*

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3): 377-404.

F. Xia and M. McCord. 2004. Improving a Statistical MT System with Automatically Learned Rewrite Patterns. *Proceedings of COLING.*

P. Xu, J. Kang, M. Ringgaard, F. Och. 2009. Using a dependency parser to improve SMT for subject-verb-object languages. *Proceedings of HLT-NAACL.*

K. Yamada and K. Knight. 2001. A Syntax-based Statistical Translation Model. *Proceedings of the 39th ACL.* Pages 523-530.

H. Zhang, L. Huang, D. Gildea and K. Knight. 2006. Synchronous binarization for machine translation. *Proceedings of the HLT-NAACL.* Pages 256-263.

B. Zhao and Y. Al-onaizan. 2008. Generalizing Local and Non-Local Word-Reordering Patterns for Syntax-Based Machine Translation. *Proceedings of EMNLP.* Pages 572-581.

A. Zollmann and A. Venugopal. 2006. Syntax augmented machine translation via chart parsing. *Proceedings of NAACL 2006 -Workshop on statistical machine translation.*

A. Zollmann, A. Venugopal, F. Och and J. Ponte. 2008. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented MT. *Proceedings of COLING.*