Using a Dependency Parser to Improve SMT for Subject-Object-Verb Languages

Peng Xu, Jaeho Kang, Michael Ringgaard and Franz Och
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043, USA
{xp,jhkang,ringgaard,och}@google.com

Abstract

We introduce a novel precedence reordering approach based on a dependency parser to statistical machine translation systems. Similar to other preprocessing reordering approaches, our method can efficiently incorporate linguistic knowledge into SMT systems without increasing the complexity of decoding. For a set of five subject-object-verb (SOV) order languages, we show significant improvements in BLEU scores when translating from English, compared to other reordering approaches, in state-of-the-art phrase-based SMT systems.

1 Introduction

Over the past ten years, statistical machine translation has seen many exciting developments. Phrase-based systems (Och, 2002; Koehn et.al., 2003; Och and Ney, 2004) advanced the machine translation field by allowing translations of word sequences (a.k.a., phrases) instead of single words. This approach has since been the state-of-the-art because of its robustness in modeling local word reordering and the existence of an efficient dynamic programming decoding algorithm.

However, when phrase-based systems are used between languages with very different word orders, such as between subject-verb-object (SVO) and subject-object-verb (SOV) languages, long distance reordering becomes one of the key weaknesses. Many reordering methods have been proposed in recent years to address this problem in different aspects.

The first class of approaches tries to explicitly model phrase reordering distances. Distance based distortion model (Och, 2002; Koehn et.al., 2003) is a simple way of modeling phrase level reordering. It penalizes non-monotonicity by applying a weight to the number of words between two source phrases corresponding to two consecutive target phrases. Later on, this model was extended to lexicalized phrase reordering (Tillmann, 2004; Koehn, et.al., 2005; Al-Onaizan and Papineni, 2006) by applying different weights to different phrases. Most recently, a hierarchical phrase reordering model (Galley and Manning, 2008) was proposed to dynamically determine phrase boundaries using efficient shift-reduce parsing. Along this line of research, discriminative reordering models based on a maximum entropy classifier (Zens and Ney, 2006; Xiong, et.al., 2006) also showed improvements over the distance based distortion model. None of these reordering models changes the word alignment step in SMT systems, therefore, they can not recover from the word alignment errors. These models are also limited by a maximum allowed reordering distance often used in decoding.

The second class of approaches puts syntactic analysis of the target language into both modeling and decoding. It has been shown that direct modeling of target language constituents movement in either constituency trees (Yamada and Knight, 2001; Galley et.al., 2006; Zollmann et.al., 2008) or dependency trees (Quirk, et.al., 2005) can result in significant improvements in translation quality for translating languages like Chinese and Arabic into English. A simpler alternative, the hierarchical phrase-based

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 245–253,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

approach (Chiang, 2005; Wu, 1997) also showed promising results for translating Chinese to English. Similar to the distance based reordering models, the syntactical or hierarchical approaches also rely on other models to get word alignments. These models typically combine machine translation decoding with chart parsing, therefore significantly increase the decoding complexity. Even though some recent work has shown great improvements in decoding efficiency for syntactical and hierarchical approaches (Huang and Chiang, 2007), they are still not as efficient as phrase-based systems, especially when higher order language models are used.

Finally, researchers have also tried to put source language syntax into reordering in machine translation. Syntactical analysis of source language can be used to deterministically reorder input sentences (Xia and McCord, 2004; Collins et.al., 2005; Wang et.al., 2007; Habash, 2007), or to provide multiple orderings as weighted options (Zhang et.al., 2007; Li et.al., 2007; Elming, 2008). In these approaches, input source sentences are reordered based on syntactic analysis and some reordering rules at preprocessing step. The reordering rules can be either manually written or automatically extracted from data. Deterministic reordering based on syntactic analysis for the input sentences provides a good way of resolving long distance reordering, without introducing complexity to the decoding process. Therefore, it can be efficiently incorporated into phrase-based systems. Furthermore, when the same preprocessing reordering is performed for the training data, we can still apply other reordering approaches, such as distance based reordering and hierarchical phrase reordering, to capture additional local reordering phenomena that are not captured by the preprocessing reordering. The work presented in this paper is largely motivated by the preprocessing reordering approaches.

In the rest of the paper, we first introduce our dependency parser based reordering approach based on the analysis of the key issues when translating SVO languages to SOV languages. Then, we show experimental results of applying this approach to phrase-based SMT systems for translating from English to five SOV languages (Korean, Japanese, Hindi, Urdu and Turkish). After showing that this approach can also be beneficial for hierarchical phrase-based systems, we will conclude the paper with future research directions.

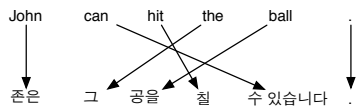

Figure 1: Example Alignment Between an English and a Korean Sentence

2 Translation between SVO and SOV Languages

In linguistics, it is possible to define a basic word order in terms of the verb (V) and its arguments, subject (S) and object (O). Among all six possible permutations, SVO and SOV are the most common. Therefore, translating between SVO and SOV languages is a very important area to study. We use English as a representative of SVO languages and Korean as a representative for SOV languages in our discussion about the word orders.

Figure 1 gives an example sentence in English and its corresponding translation in Korean, along with the alignments between the words. Assume that we split the sentences into four phrases: (John , 존은), (can hit , 칠 수 있습니다), (the ball , 그 공을) and (. , .). Since a phrase-based decoder generates the translation from left to right, the following steps need to happen when we translate from English to Korean:

- Starts from the beginning of the sentence, translates “John” to “존은”;
- Jumps to the right by two words, translates “the ball” to “그 공을”;
- Jumps to the left by four words, translates “can hit” to “칠 수 있습니다”;
- Finally, jumps to the right by two words, translates “.” to “.”.

It is clear that in order for the phrase-based decoder to successfully carry out all of the reordering steps, a very strong reordering model is required. When the sentence gets longer with more complex structure, the number of words to move over during decoding can be quite high. Imagine when we translate

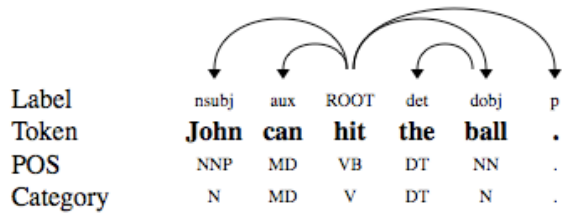

Figure 2: Dependency Parse Tree of an Example English Sentence

the sentence "**English is used as the first or second language in many countries around the world .**". The decoder needs to make a jump of 13 words in order to put the translation of "**is used**" at the end of the translation. Normally in a phrase-based decoder, very long distance reordering is not allowed because of efficiency considerations. Therefore, it is very difficult in general to translate English into Korean with proper word order.

However, knowing the dependency parse trees of the English sentences may simplify the reordering problem significantly. In the simple example in Figure 1, if we analyze the English sentence and know that "John" is the subject, "can hit" is the verb and "the ball" is the object, we can reorder the English into SOV order. The resulting sentence "John the ball can hit ." will only need monotonic translation. This motivates us to use a dependency parser for English to perform the reordering.

3 Precedence Reordering Based on a Dependency Parser

Figure 2 shows the dependency tree for the example sentence in the previous section. In this parse, the verb "hit" has four children: a subject noun "John", an auxiliary verb "can", an object noun "ball" and a punctuation ".". When transforming the sentence to SOV order, we need to move the object noun and the subtree rooted at it to the front of the head verb, but after the subject noun. We can have a simple rule to achieve this.

However, in reality, there are many possible children for a verb. These children have some relative ordering that is typically fixed for SOV languages. In order to describe this kind of ordering, we propose precedence reordering rules based on a dependency parse tree. All rules here are based English and Korean examples, but they also apply to other SOV languages, as we will show later empirically.

A precedence reordering rule is a mapping from T to a set of tuples $\{(L, W, O)\}$, where T is the part-of-speech (POS) tag of the head in a dependency parse tree node, L is a dependency label for a child node, W is a weight indicating the order of that child node and O is the type of order (either *NORMAL* or *REVERSE*). The type of order is only used when we have multiple children with the same weight, while the weight is used to determine the relative order of the children, going from largest to smallest. The weight can be any real valued number. The order type NORMAL means we preserve the original order of the children, while REVERSE means we flip the order. We reserve a special label $self$ to refer to the head node itself so that we can apply a weight to the head, too. We will call this tuple a *precedence tuple* in later discussions. In this study, we use manually created rules only.

Suppose we have a precedence rule: VB $\rightarrow$ (nsubj, 2, NORMAL), (dobj, 1, NORMAL), (self, 0, NORMAL). For the example shown in Figure 2, we would apply it to the ROOT node and result in "John the ball can hit .".

Given a set of rules, we apply them in a dependency tree recursively starting from the root node. If the POS tag of a node matches the left-hand-side of a rule, the rule is applied and the order of the sentence is changed. We go through all children of the node and get the precedence weights for them from the set of precedence tuples. If we encounter a child node that has a dependency label not listed in the set of tuples, we give it a default weight of 0 and default order type of NORMAL. The children nodes are sorted according to their weights from highest to lowest, and nodes with the same weights are ordered according to the type of order defined in the rule.

3.1 Verb Precedence Rules

Verb movement is the most important movement when translating from English (SVO) to Korean (SOV). In a dependency parse tree, a verb node can potentially have many children. For example, auxiliary and passive auxiliary verbs are often grouped together with the main verb and moved together with it. The order, however, is reversed after the movement. In the example of Figure 2, the correct Korean

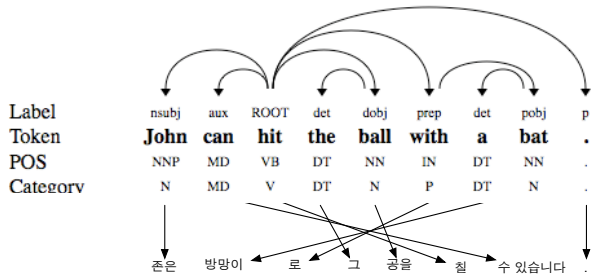

Figure 3: Dependency Parse Tree with Alignment for a Sentence with Preposition Modifier

word order is "**칠 (hit) 수 있습니다(can)** . Other categories that are in the same group are phrasal verb particle and negation.

If the verb in an English sentence has a prepositional phrase as a child, the prepositional phrase is often placed before the direct object in the Korean counterpart. As shown in Figure 3, "방망이 로" ("with a bat") is actually between "존은" ("John") and "그 공을" ("the ball").

Another common reordering phenomenon is when a verb has an adverbial clause modifier. In that case, the whole adverbial clause is moved together to be in front of the subject of the main sentence. Inside the adverbial clause, the ordering follows the same verb reordering rules, so we recursively reorder the clause.

Our verb precedence rule, as in Table 1, can cover all of the above reordering phenomena. One way to interpret this rule set is as follows: for any node whose POS tag is matches VB* (VB, VBZ, VBD, VBP, VBN, VBG), we group the children node that are phrasal verb particle (prt), auxiliary verb (aux), passive auxiliary verb (auxpass), negation (neg) and the verb itself (self) together and reverse them. This verb group is moved to the end of the sentence. We move adverbial clause modifier to the beginning of the sentence, followed by a group of noun subject (nsubj), preposition modifier and anything else not listed in the table, in their original order. Right before the verb group, we put the direct object (dobj). Note that all of the children are optional.

3.2 Adjective Precedence Rules

Similar to the verbs, adjectives can also take an auxiliary verb, a passive auxiliary verb and a negation

T	(L, W, O)
VB*	(advcl, 1, NORMAL) (nsubj, 0, NORMAL) (prep, 0, NORMAL) (dobj, -1, NORMAL) (prt, -2, REVERSE) (aux, -2, REVERSE) (auxpass, -2, REVERSE) (neg, -2, REVERSE) (self, -2, REVERSE)
JJ or JJS or JJR	(advcl, 1, NORMAL) (self, -1, NORMAL) (aux, -2, REVERSE) (auxpass, -2, REVERSE) (neg, -2, REVERSE) (cop, -2, REVERSE)
NN or NNS	(prep, 2, NORMAL) (rcmod, 1, NORMAL) (self, 0, NORMAL)
IN or TO	(pobj, 1, NORMAL) (self, -1, NORMAL)

Table 1: Precedence Rules to Reorder English to SOV Language Order (These rules were extracted manually by a bilingual speaker after looking at some text book examples in English and Korean, and the dependency parse trees of the English examples.)

as modifiers. In such cases, the change in order from English to Korean is similar to the verb rule, except that the head adjective itself should be in front of the verbs. Therefore, in our adjective precedence rule in the second panel of Table 1, we group the auxiliary verb, the passive auxiliary verb and the negation and move them together after reversing their order. They are moved to right after the head adjective, which is put after any other modifiers.

For both verb and adjective precedence rules, we also apply some heuristics to prevent excessive movements. In order to do this, we disallow any movement across punctuation and conjunctions. Therefore, for sentences like "**John hit the ball but Sam threw the ball**", the reordering result would be "**John the ball hit but Sam the ball threw**", instead of "**John the ball but Sam the ball threw hit**".

3.3 Noun and Preposition Precedence Rules

In Korean, when a noun is modified by a prepositional phrase, such as in "**the way to happiness**", the prepositional phrase is usually moved in front of the noun, resulting in "**행복 (happiness) 으로 가는 길 (to the way)**" . Similarly for relative clause modifier, it is also reordered to the front of the head noun. For preposition head node with an object modifier,

the order is the object first and the preposition last. One example is "**with a bat**" in Figure 3. It corresponds to "**방망이 (a bat) 로(with)**". We handle these types of reordering by the noun and preposition precedence rules in the third and fourth panel of Table 1.

With the rules defined in Table 1, we now show a more complex example in Figure 4. First, the ROOT node matches an adjective rule, with four children nodes labeled as (csubj, cop, advcl, p), and with precedence weights of (0, -2, 1, 0). The ROOT node itself has a weight of -1. After reordering, the sentence becomes: "**because we do n't know what the future has Living exciting is .**". Note that the whole adverbial phrase rooted at "**know**" is moved to the beginning of the sentence. After that, we see that the child node rooted at "**know**" matches a verb rule, with five children nodes labeled as (mark, nsubj, aux, neg, ccomp), with weights (0, 0, -2, -2, 0). In this case, the verb itself also has weight -2. Now we have two groups of nodes, with weight 0 and -2, respectively. The first group has a NORMAL order and the second group has a REVERSE order. After reordering, the sentence becomes: "**because we what the future has know n't do Living exciting is .**". Finally, we have another node rooted at "**has**" that matches the verb rule again. After the final reordering, we end up with the sentence: "**because we the future what has know n't do Living exciting is .**". We can see in Figure 4 that this sentence has an almost monotonic alignment with a reasonable Korean translation shown in the figure[1].

4 Related Work

As we mentioned in our introduction, there have been several studies in applying source sentence reordering using syntactical analysis for statistical machine translation. Our precedence reordering approach based on a dependency parser is motivated by those previous works, but we also distinguish from their studies in various ways.

Several approaches use syntactical analysis to provide multiple source sentence reordering options through word lattices (Zhang et.al., 2007; Li et.al., 2007; Elming, 2008). A key difference between their approaches and ours is that they do not perform reordering during training. Therefore, they would need to rely on reorder units that are likely not violating "phrase" boundaries. However, since we reorder both training and test data, our system operates in a matched condition. They also focus on either Chinese to English (Zhang et.al., 2007; Li et.al., 2007) or English to Danish (Elming, 2008), which arguably have less long distance reordering than between English and SOV languages.

Studies most similar to ours are those preprocessing reordering approaches (Xia and McCord, 2004; Collins et.al., 2005; Wang et.al., 2007; Habash, 2007). They all perform reordering during preprocessing based on either automatically extracted syntactic rules (Xia and McCord, 2004; Habash, 2007) or manually written rules (Collins et.al., 2005; Wang et.al., 2007). Compared to these approaches, our work has a few differences. First of all, we study a wide range of SOV languages using manually extracted precedence rules, not just for one language like in these studies. Second, as we will show in the next section, we compare our approach to a very strong baseline with more advanced distance based reordering model, not just the simplest distortion model. Third, our precedence reordering rules, like those in Habash, 2007, are more flexible than those other rules. Using just one verb rule, we can perform the reordering of subject, object, preposition modifier, auxiliary verb, negation and the head verb. Although we use manually written rules in this study, it is possible to learn our rules automatically from alignments, similarly to Habash, 2007. However, unlike Habash, 2007, our manually written rules handle unseen children and their order naturally because we have a default precedence weight and order type, and we do not need to match an often too specific condition, but rather just treat all children independently. Therefore, we do not need to use any backoff scheme in order to have a broad coverage. Fourth, we use dependency parse trees rather than constituency trees.

There has been some work on syntactic word order model for English to Japanese machine translation (Chang and Toutanova, 2007). In this work, a global word order model is proposed based on features including word bigram of the target sentence, displacements and POS tags on both source and tar-

[1] We could have improved the rules by using a weight of -3 for the label "*mark*", but it was not in our original set of rules.

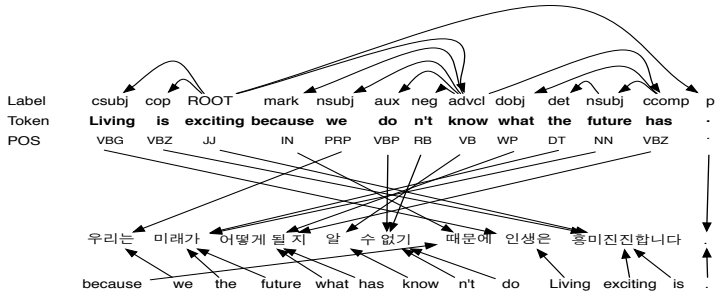

Figure 4: A Complex Reordering Example (Reordered English sentence and alignments are at the bottom.)

get sides. They build a log-linear model using these features and apply the model to re-rank N-best lists from a baseline decoder. Although we also study the reordering problem in English to Japanese translation, our approach is to incorporate the linguistically motivated reordering directly into modeling and decoding.

System	Source	Target
English→Korean	303M	267M
English→Japanese	316M	350M
English→Hindi	16M	17M
English→Urdu	17M	19M
English→Turkish	83M	76M

Table 2: Training Corpus Statistics (#words) of Systems for 5 SOV Languages

5 Experiments

We carried out all our experiments based on a state-of-the-art phrase-based statistical machine translation system. When training a system for English to any of the 5 SOV languages, the word alignment step includes 3 iterations of IBM Model-1 training and 2 iterations of HMM training. We do not use Model-4 because it is slow and it does not add much value to our systems in a pilot study. We use the standard phrase extraction algorithm (Koehn et.al., 2003) to get all phrases up to length 5. In addition to the regular distance distortion model, we incorporate a maximum entropy based lexicalized phrase reordering model (Zens and Ney, 2006) as a feature used in decoding. In this model, we use 4 reordering classes ($+1$, > 1, -1, < -1) and words from both source and target as features. For source words, we use the current aligned word, the word before the current aligned word and the next aligned word; for target words, we use the previous two words in the immediate history. Using this type of features makes it possible to directly use the maximum entropy model in the decoding process (Zens and Ney, 2006). The maximum entropy models are trained on all events extracted from training data word alignments using the LBFGS algorithm (Malouf, 2002). Overall for decoding, we use between 20 to 30 features, whose weights are optimized using MERT (Och, 2003), with an implementation based on the lattice MERT (Macherey et.al., 2008).

For parallel training data, we use an in-house collection of parallel documents. They come from various sources with a substantial portion coming from the web after using simple heuristics to identify potential document pairs. Therefore, for some documents in the training data, we do not necessarily have the exact clean translations. Table 2 shows the actual statistics about the training data for all five languages we study. For all 5 SOV languages, we use the target side of the parallel data and some more monolingual text from crawling the web to build 4-gram language models.

We also collected about 10K English sentences from the web randomly. Among them, 9.5K are used as evaluation data. Those sentences were translated by humans to all 5 SOV languages studied in this paper. Each sentence has only one reference translation. We split them into 3 subsets: *dev* contains 3,500 sentences, *test* contains 1,000 sentences and the rest of 5,000 sentences are used in a *blindtest* set. The dev set is used to perform MERT training, while the test set is used to select trained weights due to some nondeterminism of MERT training. We use IBM BLEU (Papineni et al., 2002) to evaluate

our translations and use character level BLEU for Korean and Japanese.

5.1 Preprocessing Reordering and Reordering Models

We first compare our precedence rules based preprocessing reordering with the maximum entropy based lexicalized reordering model. In Table 3, *Baseline* is our system with both a distance distortion model and the maximum entropy based lexicalized reordering model. For all results reported in this section, we used a maximum allowed reordering distance of 10. In order to see how the lexicalized reordering model performs, we also included systems with and without it (*-LR* means without it). *PR* is our proposed approach in this paper. Note that since we apply precedence reordering rules during preprocessing, we can combine this approach with any other reordering models used during decoding. The only difference is that with the precedence reordering, we would have a different phrase table and in the case of *LR*, different maximum entropy models.

In order to implement the precedence rules, we need a dependency parser. We choose to use a deterministic inductive dependency parser (Nivre and Scholz, 2004) for its efficiency and good accuracy. Our implementation of the deterministic dependency parser using maximum entropy models as the underlying classifiers achieves 87.8% labeled attachment score and 88.8% unlabeled attachment score on standard Penn Treebank evaluation.

As our results in Table 3 show, for all 5 languages, by using the precedence reordering rules as described in Table 1, we achieve significantly better BLEU scores compared to the baseline system. In the table, We use two stars ($**$) to mean that the statistical significance test using the bootstrap method (Koehn, 2004) gives an above 95% significance level when compared to the baselie. We measured the statistical significance level only for the blindtest data.

Note that for Korean and Japanese, our precedence reordering rules achieve better absolute BLEU score improvements than for Hindi, Urdu and Turkish. Since we only analyzed English and Korean sentences, it is possible that our rules are more geared toward Korean. Japanese has almost exactly the same word order as Korean, so we could assume the benefits can carry over to Japanese.

Language	System	dev	test	blind
Korean	BL	25.8	27.0	26.2
	-LR	24.7	25.6	25.1
	-LR+PR	27.3	28.3	27.5**
	+PR	27.8	28.7	27.9**
Japanese	BL	29.5	29.3	29.3
	-LR	29.2	29.0	29.0
	-LR+PR	30.3	31.0	30.6**
	+PR	30.7	31.2	31.1**
Hindi	BL	19.1	18.9	18.3
	-LR	17.4	17.1	16.4
	-LR+PR	19.6	18.8	18.7**
	+PR	19.9	18.9	18.8**
Urdu	BL	9.7	9.5	8.9
	-LR	9.1	8.6	8.2
	-LR+PR	10.0	9.6	9.6**
	+PR	10.0	9.8	9.6**
Turkish	BL	10.0	10.5	9.8
	-LR	9.1	10.0	9.0
	-LR+PR	10.5	11.0	10.3**
	+PR	10.5	10.9	10.4**

Table 3: BLEU Scores on Dev, Test and Blindtest for English to 5 SOV Languages with Various Reordering Options (BL means baseline, LR means maximum entropy based lexialized phrase reordering model, PR means precedence rules based preprocessing reordering.)

5.2 Reordering Constraints

One of our motivations of using the precedence reordering rules is that English will look like SOV languages in word order after reordering. Therefore, even monotone decoding should be able to produce better translations. To see this, we carried out a controlled experiment, using Korean as an example.

Clearly, after applying the precedence reordering rules, our English to Korean system is not sensitive to the maximum allowed reordering distance anymore. As shown in Figure 5, without the rules, the blindtest BLEU scores improve monotonically as the allowed reordering distance increases. This indicates that the order difference between English and Korean is very significant. Since smaller allowed reordering distance directly corresponds to decoding time, we can see that with the same decoding speed, our proposed approach can achieve almost 5% BLEU score improvements on blindtest set.

5.3 Preprocessing Reordering and Hierarchical Model

The hierarchical phrase-based approach has been successfully applied to several systems (Chiang,

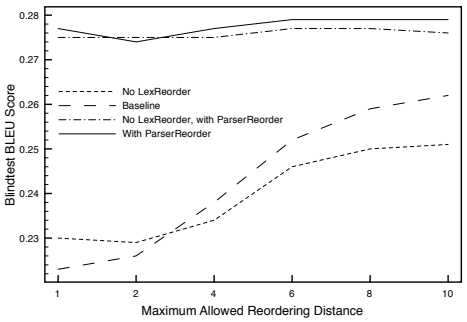

Figure 5: Blindtest BLEU Score for Different Maximum Allowed Reordering Distance for English to Korean Systems with Different Reordering Options

Language	System	dev	test	blind
Korean	PR	27.8	28.7	27.9
	Hier	27.4	27.7	27.9
	PR+Hier	28.5	29.1	28.8**
Japanese	PR	30.7	31.2	31.1**
	Hier	30.5	30.6	30.5
	PR+Hier	31.0	31.3	31.1**
Hindi	PR	19.9	18.9	18.8
	Hier	20.3	20.3	19.3
	PR+Hier	20.0	19.7	19.3
Urdu	PR	10.0	9.8	9.6
	Hier	10.4	10.3	10.0
	PR+Hier	11.2	10.7	10.7**
Turkish	PR	10.5	10.9	10.4
	Hier	11.0	11.8	10.5
	PR+Hier	11.1	11.6	10.9**

Table 4: BLEU Scores on Dev, Test and Blindtest for English to 5 SOV Languages in Hierarchical Phrase-based Systems (PR is precedence rules based preprocessing reordering, same as in Table 3, while Hier is the hierarchical system.)

2005; Zollmann et.al., 2008). Since hierarchical phrase-based systems can capture long distance reordering by using a PSCFG model, we expect it to perform well in English to SOV language systems.

We use the same training data as described in the previous sections for building hierarchical systems. The same 4-gram language models are also used for the 5 SOV languages. We adopt the SAMT package (Zollmann and Venugopal, 2006) and follow similar settings as Zollmann et.al., 2008. We allow each rule to have at most 6 items on the source side, including nonterminals and extract rules from initial phrases of maximum length 12. During decoding, we allow application of all rules of the grammar for chart items spanning up to 12 source words.

Since our precedence reordering applies at preprocessing step, we can train a hierarchical system after applying the reordering rules. When doing so, we use exactly the same settings as a regular hierarchical system. The results for both hierarchical systems and those combined with the precedence reordering are shown in Table 4, together with the best normal phrase-based systems we copy from Table 3. Here again, we mark any blindtest BLEU score that is better than the corresponding hierarchical system with confidence level above 95%. Note that the hierarchical systems can not use the maximum entropy based lexicalized phrase reordering models.

Except for Hindi, applying the precedence reordering rules in a hierarchical system can achieve statistically significant improvements over a normal hierarchical system. We conjecture that this may be because of the simplicity of our reordering rules. Other than the reordering phenomena covered by our rules in Table 1, there could be still some local or long distance reordering. Therefore, using a hierarchical phrase-based system can improve those cases. Another possible reason is that after the reordering rules apply in preprocessing, English sentences in the training data are very close to the SOV order. As a result, EM training becomes much easier and word alignment quality becomes better. Therefore, a hierarchical phrase-based system can extract better rules and hence achievesbetter translation quality.

We also point out that hierarchical phrase-based systems require a chart parsing algorithm during decoding. Compared to the efficient dynamic programming in phrase-based systems, it is much slower. This makes our approach more appealing in a realtime statistical machine translation system.

6 Conclusion

In this paper, we present a novel precedence reordering approach based on a dependency parser. We successfully applied this approach to systems translating English to 5 SOV languages: Korean, Japanese, Hindi, Urdu and Turkish. For all 5 languages, we achieve statistically significant improvements in BLEU scores over a state-of-the-art phrase-based baseline system. The amount of training data for the 5 languages varies from around 17M to more than 350M words, including some noisy data from

the web. Our proposed approach has shown to be robust and versatile. For 4 out of the 5 languages, our approach can even significantly improve over a hierarchical phrase-based baseline system. As far as we know, we are the first to show that such reordering rules benefit several SOV languages.

We believe our rules are flexible and can cover many linguistic reordering phenomena. The format of our rules also makes it possible to automatically extract rules from word aligned corpora. In the future, we plan to investigate along this direction and extend the rules to languages other than SOV.

The preprocessing reordering like ours is known to be sensitive to parser errors. Some preliminary error analysis already show that indeed some sentences suffer from parser errors. In the recent years, several studies have tried to address this issue by using a word lattice instead of one reordering as input (Zhang et.al., 2007; Li et.al., 2007; Elming, 2008). Although there is clearly room for improvements, we also feel that using one reordering during training may not be good enough either. It would be very interesting to investigate ways to have efficient procedure for training EM models and getting word alignments using word lattices on the source side of the parallel data. Along this line of research, we think some kind of tree-to-string model (Liu et.al., 2006) could be interesting directions to pursue.

References

Yaser Al-Onaizan and Kishore Papineni 2006. Distortion Models for Statistical Machine Translation *In Proceedings of ACL*

Pi-Chuan Chang and Kristina Toutanova 2007. A Discriminative Syntactic Word Order Model for Machine Translation *In Proceedings of ACL*

David Chiang 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation *In Proceedings of ACL*

Michael Collins, Philipp Koehn and Ivona Kucerova 2005. Clause Restructuring for Statistical Machine Translation *In Proceedings of ACL*

Jakob Elming 2008. Syntactic Reordering Integrated with Phrase-based SMT *In Proceedings of COLING*

Michel Galley and Christopher D. Manning 2008. A Simple and Effective Hierarchical Phrase Reordering Model *In Proceedings of EMNLP*

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang and Ignacio Thayer 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models *In Proceedings of COLING-ACL*

Nizar Habash 2007. Syntactic Preprocessing for Statistical Machine Translation *In Proceedings of 11th MT Summit*

Liang Huang and David Chiang 2007. Forest Rescoring: Faster Decoding with Integrated Language Models, *In Proceedings of ACL*

Philipp Koehn 2004. Statistical Significance Tests for Machine Translation Evaluation *In Proceedings of EMNLP*

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne and David Talbot 2005. Edinborgh System Description for the 2005 IWSLT Speech Translation Evaluation *In International Workshop on Spoken Language Translation*

Philipp Koehn, Franz J. Och and Daniel Marcu 2003. Statistical Phrase-based Translation, *In Proceedings of HLT-NAACL*

Chi-Ho Li, Dongdong Zhang, Mu Li, Ming Zhou, Minghui Li and Yi Guan 2007. A Probabilistic Approach to Syntax-based Reordering for Statistical Machine Translation, *In Proceedings of ACL*

Yang Liu, Qun Liu and Shouxun Lin 2006. Tree-to-string Alignment Template for Statistical Machine Translation, *In Proceedings of COLING-ACL*

Wolfgang Macherey, Franz J. Och, Ignacio Thayer and Jakob Uszkoreit 2008. Lattice-based Minimum Error Rate Training for Statistical Machine Translation *In Proceedings of EMNLP*

Robert Malouf 2002. A comparison of algorithms for maximum entropy parameter estimation *In Proceedings of the Sixth Workshop on Computational Language Learning (CoNLL-2002)*

Joakim Nivre and Mario Scholz 2004. Deterministic Dependency Parsing for English Text. *In Proceedings of COLING*

Franz J. Och 2002. Statistical Machine Translation: From Single Word Models to Alignment Template Ph.D. Thesis, RWTH Aachen, Germany

Franz J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *In Proceedings of ACL*

Franz J. Och and Hermann Ney 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics, 30:417-449*

Kishore Papineni, Roukos, Salim et al. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. *In Proceedings of ACL*

Chris Quirk, Arul Menezes and Colin Cherry 2005. Dependency Tree Translation: Syntactically Informed Phrasal SMT *In Proceedings of ACL*

Christoph Tillmann 2004. A Block Orientation Model for Statistical Machine Translation *In Proceedings of HLT-NAACL*

Chao Wang, Michael Collins and Philipp Koehn 2007. Chinese Syntactic Reordering for Statistical Machine Translation *In Proceedings of EMNLP-CoNLL*

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

Fei Xia and Michael McCord 2004. Improving a Statistical MT System with Automatically Learned Rewrite Patterns *In Proceedings of COLING*

Deyi Xiong, Qun Liu and Shouxun Lin 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation *In Proceedings of COLING-ACL*

Kenji Yamada and Kevin Knight 2001. A Syntax-based Statistical Translation Model *In Proceedings of ACL*

Yuqi Zhang, Richard Zens and Hermann Ney 2007. Improve Chunk-level Reordering for Statistical Machine Translation *In Proceedings of IWSLT*

Richard Zens and Hermann Ney 2006. Discriminative Reordering Models for Statistical Machine Translation *In Proceedings of the Workshop on Statistical Machine Translation, HLT-NAACL* pages 55-63

Andreas Zollmann and Ashish Venugopal 2006. Syntax Augmented Machine Translation via Chart Parsing *In Proceedings of NAACL 2006 - Workshop on Statistical Machine Translation*

Andreas Zollmann, Ashish Venugopal, Franz Och and Jay Ponte 2008. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT *In Proceedings of COLING*