Improved Phrase-based SMT with Syntactic Reordering Patterns Learned from Lattice Scoring

Jie Jiang, Jinhua Du, Andy Way
CNGL, School of Computing, Dublin City University,
Glasnevin, Dublin 9, Ireland
{jjiang,jdu,away}@computing.dcu.ie

Abstract

In this paper, we present a novel approach to incorporate source-side syntactic reordering patterns into phrase-based SMT. The main contribution of this work is to use the lattice scoring approach to exploit and utilize reordering information that is favoured by the baseline PBSMT system. By referring to the parse trees of the training corpus, we represent the observed reorderings with source-side syntactic patterns. The extracted patterns are then used to convert the parsed inputs into word lattices, which contain both the original source sentences and their potential reorderings. Weights of the word lattices are estimated from the observations of the syntactic reordering patterns in the training corpus. Finally, the PBSMT system is tuned and tested on the generated word lattices to show the benefits of adding potential source-side reorderings in the inputs. We confirmed the effectiveness of our proposed method on a medium-sized corpus for Chinese-English machine translation task. Our method outperformed the baseline system by 1.67% relative on a randomly selected testset and 8.56% relative on the NIST 2008 testset in terms of BLEU score.

1 Introduction

To take consideration of reordering problem between different language pairs, phrase-based statistical machine translation (PBSMT) systems (Koehn et al., 2003) incorporate two different of methods: 1) learning phrase pairs with different word orders in the source and target sentences; 2) attempting potential target phrase orders during the decoding phase, and penalizing potential phrase orders using both distance-based and lexical reordering models. However, for some language pairs, this model is not powerful enough to capture the word order differences between the source and target sentences. To tackle this problem, previous studies (Wang et al., 2007a; Chang et al., 2009a) showed that syntactic reorderings can benefit state-of-the-art PBSMT systems by handling systematic differences in word order between language pairs. From their conclusions, for the Chinese-English task, syntactic reorderings can greatly improve the performance by explicitly modeling the structural differences between this language pair.

Interestingly, lots of work has been reported on syntactic reorderings and similar conclusions have been drawn from them. These methods can be roughly divided into two main categories (Elming, 2008): the deterministic reordering approach and the non-deterministic reordering approach.

For the deterministic approach, syntactic reorderings take place outside the PBSMT system, and the corresponding PBSMT systems only deal with the reordered source sentences. In this approach, syntactic reorderings can be performed by manually created rules (Collins et al., 2005; Wang et al., 2007a), or by rules extracted automatically from parse trees (Collins et al., 2005; Habash, 2007). For some typical syntactic structures (e.g. *DE* construction in Chinese), classifiers (Chang et al., 2009b; Du et al., 2010) are built to carry out source reorderings.

For the non-deterministic approach, both the original and reordered source sentences are fed into the PBSMT decoders, and the decisions are left to the decoders to choose the most appropriate one. (Crego et al., 2007) used syntactic structures to reorder the input into word lattices for N-gram-based Statistical Machine Translation. (Zhang et al., 2007a; Zhang et al., 2007b) employed chunks and POS tags to extract reordering rules, language models and reordering models are also used to weight the generated word lattices. Weighted n-best lists generated from rules are also used in (Li et al., 2007) for input into the decoders, while the rules are created from a syntactic parser. On the other hand, using the syntactic rules to score the output word order is adopted by (Elming, 2008; Elming, 2009), both on English-Danish and English-Arabic tasks, which confirmed the effectiveness of syntactic reorderings for distant language pairs. Another related pieces of work applies syntactic reordering information extracted from phrase orientation classifiers as an extra feature in PBSMT systems (Chang et al., 2009b) for a Chinese-English task.

However, rewriting the source sentence cannot be undone by the decoders (Al-Onaizan et al., 2006), which makes the deterministic approach less flexible than the non-deterministic one. Nevertheless, for the non-deterministic approach, most of the work relies on the syntactic information (cf. parse tree, chunks, POS tags) but never addresses which kind of rules are favoured by the decoders in SMT systems. Accordingly, the final systems might not benefit from many of the reordering rules.

In this paper, we adopt the lattice scoring approach proposed in (Jiang et al., 2010) to discover reorderings contained in phrase alignments that are favoured by a baseline PBSMT system. Given this, the central idea of this work is to feed these reorders back to the baseline PBSMT system with optional reordering information on the source-side, and let the decoder choose better reorderings according to our inputs. To accomplish this, syntactic reordering patterns on the source side are used to represent the potential reorderings from the lattice scoring outputs. However, these patterns are also used to transform the baseline inputs into word lattices to carry potential reorderings that are useful for PBSMT decoders.

The other main contributions of this work are:

- Syntactic reordering patterns are automatically extracted from lattice scoring outputs which show the preferences of the baseline PBSMT system, rather than heuristic rules.

- Our method is seamlessly incorporated with existing distance-based and lexical reordering models, as the potential reorderings are constructed on the source-side with word lattices.

The rest of this paper is organized as follows: In section 2 we give a brief overview of the lattice scoring approach for PBSMT systems, as well as the generated phrase alignments. In section 3 we discuss the extraction process of syntactic reordering patterns from phrase aligned sentences in the training corpus. Then in section 4 we present the way to transform inputs into word lattices with syntactic reordering patterns. After that, we present our experiments setup and results, as well as the discussions in section 5. Finally, we give the conclusion and future work in section 6.

2 Lattice scoring for phrase alignments

The lattice scoring approach was previously proposed in (Jiang et al., 2010) for data cleaning. The idea of that work is to utilize word alignments to perform approximated decoding on the training corpus, thus to calculate BLEU (Papineni et al., 2002) scores from the decoding results which are subsequently used to filter out low score sentences pairs. The lattice scoring procedure contains the following steps: 1) Train an initial PBSMT model on the given corpus; 2) Collect anchor pairs containing both the source and target side phrase positions from word alignments generated from the training phase; 3) Build source-side lattices from the anchor pairs and the translation model; 4) Expand and search on the source-side lattices to obtain an approximated decoding result; 5) Calculate BLEU scores on the training set and filter sentence pairs with lower scores. Step 5 is only useful for data cleaning, but steps 1-4 can be used to extract reordering information in this paper.

By taking the lattice scoring steps above, it is interesting that in step 4, not only the approximated decoding results are obtained, but also its corresponding phrase alignments can be tracked. That is because the source-side lattices built in step 3 are come from anchor pairs, so each edge in the lattices contains both the source and target side phrase positions. Once the best paths are searched for in step 4, we can obtain sequences of phrase alignments between source and target side sentences. A sample of the phrase alignments generated from lattice scoring is illustrated in Figure 1.

In Figure 1, the source sentence (Chinese) is shown on the right hand side of the alignments and the target sentence (English) is on the bottom. Note that different from word alignments, elements of the alignments in Figure 1 are phrases, and the alignment points in the figure indicates the relationship between source and target phrases which are segmented from the lattice scoring approach. Not all the phrases have alignment points because implicit edges are chosen during the search phase of lattice scoring (Jiang et al., 2010).

Rather than using word alignments (Crego et al., 2007) or phrase alignments from heuristic rules (Xia et al., 2004), we use phrase alignments generated from lattice scoring, because this incorporates the PBSMT model to score potential phrase segmentations and alignments, and only those phrase segmentations and alignments have a higher model score are selected, while unlikely reorderings from word alignments for PBSMT model are filtered before pattern extraction, hence we can obtain better reordering patterns after that has taken place. In the following section, we use this information to extract reorderings, which also indicate higher model scores from the PBSMT model.

3 Reordering patterns

In the last section, we obtained phrase alignments from the lattice scoring procedure. From the alignment points, the reordering is shown in the non-monotonic region of Figure 1, i.e. between source words 8-13 and target words 7-12, there is a non-monotonic alignment region. By comparing source and target texts within this region, there is a structural word order difference between Chinese and English, which is specified as the *DE* construction in (Chang et al., 2009a; Du et al., 2010). However, in this paper, instead of dealing with a specified reordering structure for one language pair, we aim at using reordering patterns to discover *any* kind of potential source-side syntactic reordering patterns from phrase alignments.

3.1 Reordering regions extraction

Unlike previous work in (Wang et al., 2007a; Chang et al., 2009a) which is carried out directly from parse trees in a top-down approach, our work aims at utilizing reorder information in phrase alignments. Accordingly, we use a bottom-up approach similar to (Xia et al., 2004; Crego et al., 2007) in this paper. We start by locating the reordering regions in the non-monotonic areas in the phrase alignments, and thereafter use syntactic patterns to describe such reorderings.

As is shown in Figure 1, to accomplish the same phrase orders on both source and target sides, supposed we retain the target sentence orders and try to adjust the phrase order on the source-side, one possible reordering operation is to swap the regions A and B on the source-side, where regions A and B contain source words 8-10 and 11-13 respectively. In this paper, reordering regions A and B indicating swapping operations on the source side are only considered as potential source-side reorderings, thus regions AB imply (1):

$$AB \Rightarrow BA \tag{1}$$

on the source-side word sequences.

For each non-monotonic area in the phrase alignments, all its sub-areas are attempted to extract reordering regions A and B, and each of them are fed into the pattern extraction process. The reason for doing this is the phrase alignments from lattice scoring cannot always be perfectly matched with parse trees (specified in the next section), and sometimes reordering regions from sub-areas can produce more meaningful patterns.

3.2 Reordering patterns from parse trees

Reordering regions AB extracted from the non-monotonic areas of the phrase alignments cannot be directly used to perform source-side reorderings, because they are just sequences of source-side words. To extract useful information from them, we map reordering regions onto parse trees to obtain syntactic reordering patterns, similar to previous work in (Xia

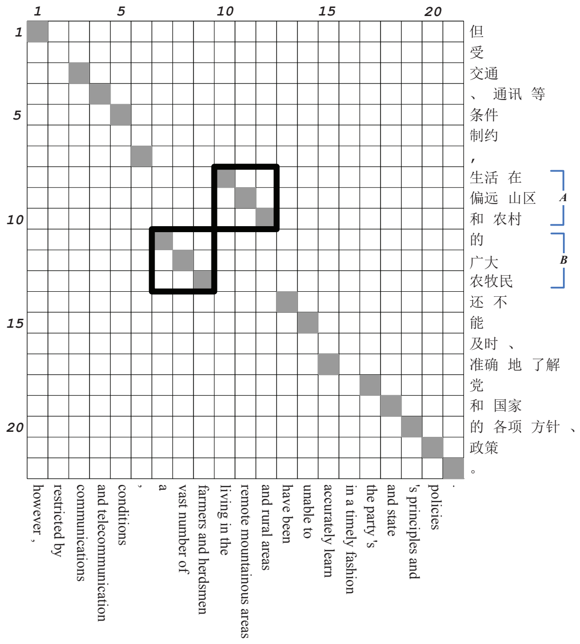

Figure 1: Phrase alignments and reorderings

et al., 2004; Crego et al., 2007). However, in this paper, the Chinese Treebank (Xue, 2005) tag set is used, and the aim is to extract appropriate patterns from them for reordering type AB in formula (1). The following steps are taken to accomplish this:

1. Parse the source side sentences into parse trees. We use the Berkeley parser (Petrov, 2006) for parsing purposes, and all parse trees are right-binarized to generate simpler tree structures for pattern extraction.

2. For each of the reordering regions AB extracted in Section 3.1, denote N_A as the node set corresponding with the words in region A and N_B for region B. The objective is to find a **minimum** treelet T of the whole parse tree, where T satisfies the following two criteria: 1) there must exist a path from each node in $N_A \cup N_B$ to the root node of T; 2) each **leaf** node of T cannot be the ancestor of nodes in both N_A and N_B (which means each leaf node can only be the ancestor of nodes in N_A, N_B, or none of them).

3. Transform T into reordering patterns P by traversing it in preorder, and at the same time, label all the leaf nodes of T with A or B as reorder options, which indicates that the descendants of nodes labeled with A are meant to swap with those of nodes labeled with B.

Instead of using *subtrees*, we use *treelets* to refer the located parse tree substructures, since treelets do not necessarily go down to leaf nodes.

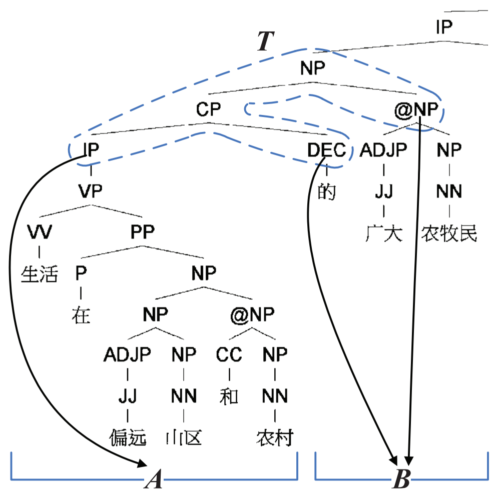

Figure 2: Reordering pattern extraction

The extraction process is illustrated in Figure 2. Part of the source-side parse tree is shown for the pattern extraction process of region AB in Figure 1. The parse tree is binarized and the symbol @ is used to indicate the extra tags generated in tree binarization (for example, $@NP$ in Figure 2).

As depicted in Figure 2, tree T (surrounded by dashed lines) is the minimum treelet of the parse tree that satisfies the two criteria in step 2 of section 3.2. Note also that the leaf nodes of T are labeled by A or B according to their descendants, e.g. IP (in region A) is labeled by A, DEC and $@NP$ (in region B) are labeled by B. After the tree T is found, we convert it into a syntactic reordering pattern by traversing it in preorder. At the same time, we collect leaf nodes labeled A or B into reordering node sequences L_A or L_B respectively to record the reordering operations. Futhermore, in order to generate larger sets of patterns, we do not distinguish tags generated in the parse tree binarization process with the original ones, which means that we treat $@NP$ and NP as the same tag. Thus, we obtain a syntactic reordering pattern P from T as in (2):

$$P = \{NP\ (CP\ (IP\ DEC)\ NP)|O = \{L_A, L_B\}\} \tag{2}$$

where the first part of P is the NP with its tree structure, and the second part O indicates the reordering scheme, which implies that source words corresponding with descendants of L_A are supposed to swap with those of L_B.

3.3 Context tags in reordering patterns

As specified at the end of section 3.1, phrase alignments cannot always be perfectly matched with parse tree topologies, especially when all sub parts of non-monotonic areas of phrase alignment are considered as potential AB reordering regions. Figure 3 illustrates this situation where there is no matched treelet for reorder regions AB.

In this case, we expand AB to the right and/or the left side with a limited number of words to find a minimum treelet which is specified in step 2 of section 3.2. In the figure, the tree node with tag P is selected when expanding region A one word to the left, such that the corresponding treelet T can be obtained. Note that in this situation, a minimum number of ancestors of expanded tree nodes are kept

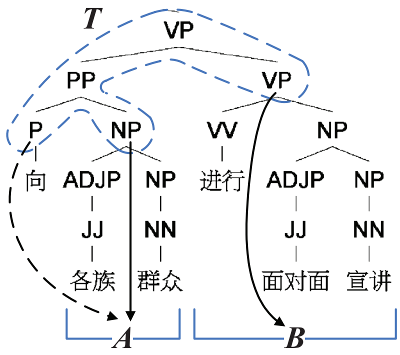

Figure 3: Context tag in pattern extraction

in T but they are assigned same labels as those from which they have been expanded, e.g. in Figure 3, the node with tag P (not in region A) is expanded from region A, so it is kept in T but labeled with A (linked with dashed arrow in the figure).

We consider expanded tree nodes as the context of syntactic reordering patterns, since they are siblings of the ancestors of word nodes in reordering regions AB. If their structure is frequently observed in the corpus, there is a greater chance that structural differences exist between source and target languages. For example, treelet T with the P tag in Figure 3 is the reordering when VP occurs with a PP modifier, which is specified in (Wang et al., 2007a). Thus, the syntactic reordering pattern for Figure 3 is as in (3):

$$P = \{VP\ (PP\ (P\ NP)\ VP)|O\} \tag{3}$$

However, the previous steps tend to generated duplicate reordering patterns because each sub-area of the non-monotonic phrase alignments are attempted and node expanding is carried out. To remove the duplications, a merge operation is carried out as follows: suppose treelets T_1 and T_2 are extracted from the same sentences while sharing the same root symbol, if T_1 is also a treelet of T_2 and their reordering regions AB overlap, then T_2 is merged into T_1. However, not all the reordering regions will generated a pattern because some of them will not have a corresponding minimum treelet.

3.4 Pattern weights estimation

Syntactic reordering patterns are extracted from non-monotonic phrase alignments. However, in the training corpus, there is not always a reordering where a treelet matches a pattern. To describe the chance of reordering p_{reo} when a treelet is matched with a pattern P, we count the occurrences of P in the training corpus, and also count the number of reorderings where there is a reordering indicated by P, and estimate it as in (4):

$$p_{reo}(P) = \frac{count\{P \;\; with \;\; reordering\}}{count\{P \;\; observed\}} \qquad (4)$$

By contrast, one syntactic pattern P usually contains more than one reordering scheme from different reordering regions and parse trees, so we assign each reordering scheme O (specified in formula (2)) with a weight as in (5):

$$w(O, P) = \frac{count\{reordering \;\; O \;\; in \;\; P\}}{count\{P \;\; with \;\; reordering\}} \qquad (5)$$

Thus, generally, a syntactic reordering pattern is expressed as in (6):

$$P = \{tree \mid p_{reo} \mid O_1, w_1, \cdots, O_n, w_n\} \qquad (6)$$

where $tree$ indicates the tree structures of the pattern, which have a reordering probability p_{reo}, and also contain n reordering schemes with weights.

4 Applying syntactic reordering patterns

Similar to (Crego et al., 2007; Zhang et al., 2007a; Zhang et al., 2007b), we use extracted patterns to transform source-side sentences into word lattices. Sentences in both the development and test sets are transformed into word lattices for potential reorderings, where a tree structure of a pattern is a treelet of a source-side parse tree.

A toy example is depicted in Figure 4. In the figure, treelet T' of the source-side parse tree is matched with a pattern. Leaf nodes $\{a_1, \cdots a_m\} \in L_A$ of T' have a span from $\{w_1, \cdots, w_p\}$ in the source sentence, while $\{b_1, \cdots, b_n\} \in L_B$ have a span from $\{v_1, \cdots, v_q\}$. Applying the reordering operation in formula (1), we add an edge from the start of w_1 to the end of v_q by swapping $\{w_1, \cdots, w_p\}$ with $\{v_1, \cdots, v_q\}$.

For each source sentence, all matched patterns are sorted by weights p_{reo} in formula (6), and a predefined number of reorderings are applied to generate lattice. For each node in the lattice with an

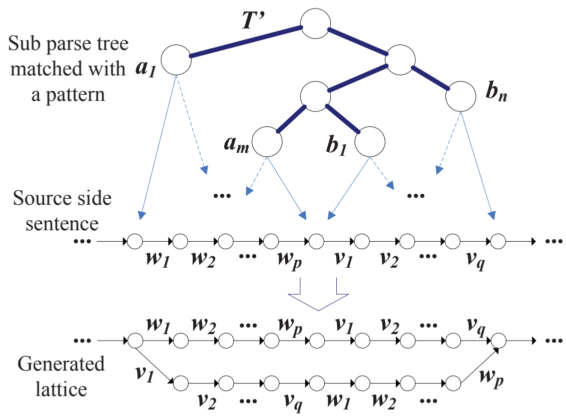

Figure 4: Applying patterns

initial edge E_0 coming from the original source sentence, if there are outgoing edges generated from patterns $\{P_1, \cdots, P_i, \cdots, P_k\}$, the weights for E_0 are defined as in (7):

$$w(E_0) = \alpha + \sum_{i=1}^{k} \{\frac{(1-\alpha)}{k} * \{1 - p_{reo}(P_i)\}\} \qquad (7)$$

where α is the base probability to avoid E_0 being equal to zero, and $p_{reo}(P_i)$ is the weight of pattern defined in formula (4). By contrast, suppose that P_i has r reordering schemes corresponding with $\{E_s, \cdots, E_{s+r-1}\}$, then weight for E_j is defined as in (8):

$$w(E_j) = \frac{(1-\alpha)}{k} * p_{reo}(P_i) * \frac{w_{s-j+1}(P_i)}{\sum_{t=1}^{r} w_t(P_i)} \qquad (8)$$

where $s <= j < s + r$, and $w_t(P_i)$ is the reordering scheme weight defined in formula (5). Here we suppose equal probabilities for all possible reorderings which start with a same lattice node.

5 Experiments

The experiments are conducted on a medium-sized corpus for Chinese-English task. The training data is the FBIS corpus, which is a multilingual paragraph-aligned corpus with LDC resource number LDC2003E14, and we use the Champollion aligner (Ma, 2006) to perform sentence alignment to obtain 256,911 sentence pairs. We randomly selected 2,000 pairs for devset and another 2,000 pairs for test set, which is referred as FBIS set in this paper. The rest of the data is used as the training set.

Evaluation results are reported on two different sets: FBIS set and the NIST 2008 test data. For FBIS set, only one reference translation is avaible for both devset and testset. For NIST data, we use the NIST 2005 test set which includes 1,082 sentences as the devset, while the NIST 2008 set is used as the test set with 1,357 sentences. In both devset and testset of NIST data, there are four reference translations for each of the sentences.

Moses (Koehn et al., 2007) is used as a baseline. Word alignment is performed with GIZA++[1] and is refined with the "grow-diag-final" method (Koehn et al., 2005), while tuning is performed with Minimum error rate training (MERT) (Och, 2003). We also use SRILM[2] to build 5-gram language models for all the experiments with modified Kneser-Ney smoothing (Kneser & Ney, 1995).

The pattern extraction experiments and the results are reported in the following subsections.

5.1 Pattern extraction

The lattice scoring approach is performed in a similar manners to that of (Jiang et al., 2010). We use the same baseline system as specified above to accomplish the lattice scoring procedure. However, instead of NIST data, the initial PBSMT system is tuned with FBIS devset to obtain weights for lattice scoring. After that, we collect anchor pairs and build source-side lattices based on the word alignments generated in the training phase. Then Viterbi search is carried out to generate phrase alignments.

From the training corpus, 48,285 syntactic reordering patterns with a total of 57,861 reordering schemes are extracted from phrase alignments. The average number of non-terminals in all patterns is 11.02. However, for reason of computational efficiency, we pruned any patterns with non-terminal numbers less than 3 and more than 9. This leaves 18,169 remaining syntactic reordering patterns with 22,850 reordering schemes, with a average number of 7.6 non-terminals.

[1]http://fjoch.com/GIZA++.html
[2]http://www.speech.sri.com/projects/srilm/

5.2 Lattice building

We apply the pruned syntactic reordering patterns to both the devset and testset, and convert source sentences of both sets into word lattices. However, the lattices size increases dramatically with respect to the number of applied patterns. To guarantee manageable word lattice inputs for the Moses decoder, we also constrain the generating process of word lattices with empirical parameters: for each source sentence, the maximum number of reordering schemes is set to 30, and the maximum span of a pattern is set to 30.

To calculate the weights of word lattices, we set the base probability in formula (7) and (8) to be 0.05. The generated word lattices of the devset and the testset are fed into Moses for tuning and evaluation respectively. No extra training steps are required.

The built-in reordering models (distance-based and lexical reordering) of Moses are also enabled while dealing with word lattice inputs, and their weights in the log-linear model (including lattice input weights) are tuned at the same time.

5.3 Results on FBIS set

To compare with the built-in reordering models of Moses, we set the distortion-limit (DL) parameter of Moses to be $\{0, 6, 10, 12\}$, and the evaluation results of the testset on FBIS data are shown in Table 1.

System	DL	BLEU	NIST	METEOR
Baseline	0	22.32	6.45	52.51
	6	23.67	6.63	54.07
	10	24.52	6.66	54.04
	<u>12</u>	<u>24.57</u>	<u>6.69</u>	<u>54.31</u>
Lattices	0	**23.92**	**6.60**	**54.30**
	6	**24.57**	**6.68**	**54.64**
	<u>10</u>	<u>**24.98**</u>	<u>**6.71**</u>	<u>**54.67**</u>
	12	**24.84**	6.69	**54.65**

Table 1: Results on FBIS testset

As shown in Table 1, for BLEU, NIST and METEOR scores, the best performance of the baseline system is achieved withdistortion limit 12 (underlined), and the best peroformance of our syntactic reordering method is obtained with ditortion limit 10 (underlined). Our method outperformed the baseline by 0.41 (1.67% relative) BLEU points, 0.02

(0.30% relative) NIST points and 0.36 (0.66% relative) METEOR points respectively. The comparison between the baseline system and our method with the same distortion limits shows that the improvments are consistent for all distortion limits (scores with bold face) except the NIST score with distortion limit 12. However, these results still confirm our proposed method on the FBIS data.

5.4 Results on NIST set

As in the last section, we also adopt serveral distortion limit parameters, and report NIST evaluation results in Table 2.

System	DL	BLEU	NIST	METEOR
Baseline	0	14.43	5.75	45.03
	6	15.61	5.88	45.75
	10	15.73	5.78	45.27
	12	15.89	6.16	45.88
Lattices	0	**16.77**	**6.54**	**47.16**
	6	**17.25**	**6.67**	**47.65**
	10	**17.15**	**6.64**	**47.78**
	12	**16.88**	**6.56**	**47.17**

Table 2: Results on NIST testset

From Table 2, the best performance of the baseline system is achieved with ditortion limit 12 (underlined), while for our method, the best BLEU and NIST scores are obtained with distortion limit 6 (underlined), and the best METEOR score is accomplished with distortion limit 10 (underlined). Our proposed method significally outperformed the baseline system by 1.36 (8.56% relative) BLEU points, 0.51 (8.28% relative) NIST points and 1.90 (4.14% relative) METEOR points respectively. Similary, the comparison between the baseline system and our moethod with the same distortion limits demostrates that the improvments are also consistent for all distortion limits (scores with bold face). These results indicate the effectiveness of the syntactic reordering model on the NIST 08 data for our medium-sized corpus.

5.5 Discussion

From the results shown in the previous sections, we found that our method can benefit the baseline PBSMT system with its built-in reordering models. But we observed that with a larger distortion limit, the improvements become lesser singnificant. This isbecause with larger distortion limit of PBSMT, the baseline system can try longer reorderings, while our method has a restriction on the range of the reordering patterns. In this case, the number of reorderings that are considered by our method but not tried by the baseline systems become lesser. Thus the improvements of our method become smaller.

However, we can still improve the system by 0.9 (3.8% relative) and 1.64 (10.5% relative) BLEU points for the two testset with distortion limit 6, which is the default setting of Moses. And with all distortion limits, our method can benefit the baseline system for different automatic evalutaion metircs. This indicates that our method can provide extra reordering capabilities for the built-in reordering models of PBSMT.

We also compare system performance with respect to the distortion limit parameter of Moses in Figure 5 and 6 for FBIS testset and NIST testset respectively. In the figure, for each of the three automatic evaluation metrics, the baseline system performance tends to have a better results with a larger distortion limit, while for lattice inputs, medium distortion limits lead to better performance. This indicates that, with lattice inputs which have already considered potential reordering on the source side, large distortion limits do not further benefit the SMT system. From this point of view, it also indicates that long range reordering might be captured well by syntactic reordering. By contrast, short range reorderings are supposed to be handled well by distance-based and lexical reordering models. Thus, for our proposed syntactic reordering enhanced system, a medium distortion limit should be preferred. However, in the experiments, our method do provides consistent improvemnts for all distortion limits.

6 Conclusion and future work

A novel approach of syntactic reorderings for PBSMT systems is studied in this paper. It aims at a bottom-up approach to extract syntactic reordering patterns from phrase alignments generated via lattice scoring, which indicates reorders favoured by the baseline system. Word lattices are used to represent potential source-side reorderings. Pattern

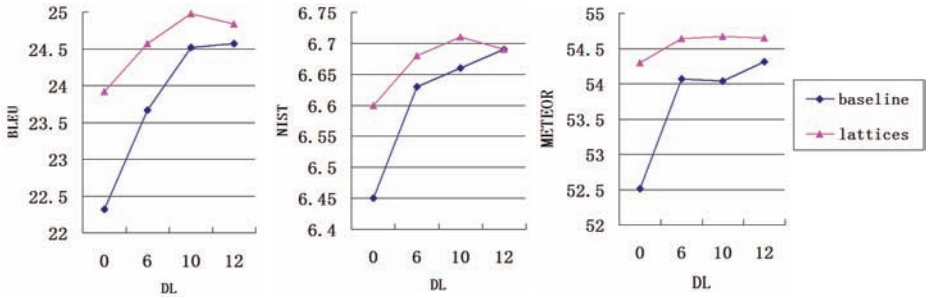

Figure 5: Score comparison on FBIS testset (DL = distortion limit)

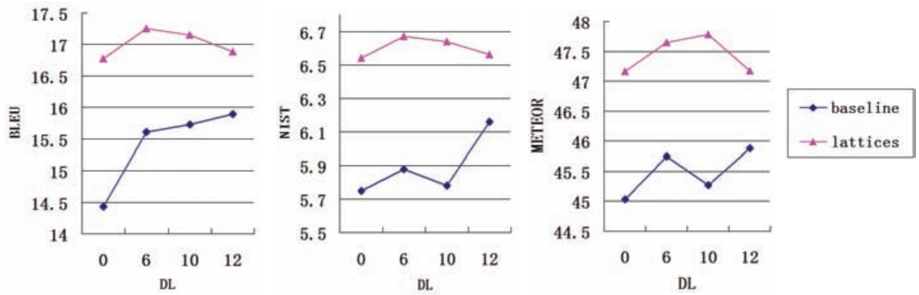

Figure 6: Score comparison on NIST testset (DL = distortion limit)

weights are estimated from the training corpus and are used to determine the edge weights in the word lattices. The proposed approach is integrated with existing distance-based and lexical reordering models, and their weights in a log-linear model are tuned with MERT. Experiments on a medium-sized corpus showed consistents improvements with all distortion limits. Compared with the baseline system, we obtained improvements of 1.67% relative on a randomly selected testset and 8.56% relative on the NIST 2008 testset in terms of BLEU score.

In the future, we plan to carry out experiments on large corpus. Furthermore, a large range of reordering types will be examined to extract more fine-grained patterns. We will also try different methods of binarizing parse trees (Wang et al., 2007b) to improve the pattern extraction process still further.

Acknowledgments

References

Yaser Al-Onaizan and Kishore Papineni 2006. Distortion models for statistical machine translation. *Coling-ACL 2006: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529-536, Sydney, Australia.

Pi-Chuan Chang, Dan Jurafsky, and Christopher D.Manning 2009a. Disambiguating DE for Chinese-English machine translation. *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 215-223, Athens, Greece.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009b. Discriminative reordering with Chinese grammatical features. *Proceedings of SSST-3: Third Workshop on Syntax and Structure in Statistical Translation*, pages 51-59, Boulder, CO.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. *ACL-2005: 43rd Annual meeting of the Association for Computational Linguistics*, pages 531-540, University of Michigan, Ann Arbor, MI.

Josep M. Crego, and José B. Mariño. 2007. Syntax-enhanced N-gram-based SMT. *MT Summit XI*, pages 111-118, Copenhagen, Denmark.

Jinhua Du and Andy Way. 2010. The Impact of Source-Side Syntactic Reordering on Hierarchical Phrase-based SMT. *EAMT 2010: 14th Annual Conference of the European Association for Machine Translation*, Saint-Raphaël, France.

Jakob Elming. 2008. Syntactic reordering integrated with phrase-based SMT. *Coling 2008: 22nd International Conference on Computational Linguistics, Pro-*

ceedings of the conference, pages 209-216, Manchester, UK.

Jakob Elming, and Nizar Habash. 2009. Syntactic reordering for English-Arabic phrase-based machine translation. *Proceedings of the EACL 2009 Workhop on Computational Approaches to Semitic Languages*, pages 69-77, Athens, Greece.

Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. *MT Summit XI*, pages 215-222, Copenhagen, Denmark.

Jie Jiang, Andy Way, Julie Carson-Berndsen. 2010. Lattice Score-Based Data Cleaning For Phrase-Based Statistical Machine Translation. *EAMT 2010: 14th Annual Conference of the European Association for Machine Translation*, Saint-Raphaël, France.

Reinhard Kneser, Hermann Ney. 1995. Improved Backing-Off for M-Gram Language Modeling. *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 181-184, Detroit, MI.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. *In Proceedings of the Human Language Techonology Conference and the North American Association for Computational Linguistics*, pages 127-133, Edmonton, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. *International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation [IWSLT 2005]*, Pittsburgh, PA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *ACL 2007: proceedings of demo and poster sessions*, pp. 177-180, Prague, Czech Republic.

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. *ACL 2007: proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 720-727, Prague, Czech Republic.

Xiaoyi Ma. 2006. Champollion: A Robust Parallel Text Sentence Aligner. *LREC 2006: Fifth International Conference on Language Resources and Evaluation*, pp.489-492, Genova, Italy.

Franz Josef Och, Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, pp. 295-302, Philadelphia, PA.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *ACL-2003: 41st Annual meeting of the Association for Computational Linguistics*, pp. 160-167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method For Automatic Evaluation of Machine Translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, pp.311-318, Philadelphia, PA.

Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. *Coling-ACL 2006: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433-440, Sydney, Australia.

Chao Wang, Michael Collins, and Philipp Koehn. 2007a. Chinese syntactic reordering for statistical machine translation. *EMNLP-CoNLL-2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 737-745, Prague, Czech Republic.

Wei Wang, Kevin Knight and Daniel Marcu. 2007b. Binarizing Syntax Trees to Improve Syntax-Based Machine Translation Accuracy. *EMNLP-CoNLL-2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 737-745, Prague, Czech Republic.

Fei Xia, and Michael McCord 2004. Improving a statistical MT system with automatically learned rewrite patterns. *Coling 2004: 20th International Conference on Computational Linguistics*, pages 508-514, University of Geneva, Switzerland.

Nianwen Xue, Fei Xia, Fu-dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2), pages 207-238.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP*, pages 333-341, Suntec, Singapore.

Yuqi Zhang, Richard Zens, and Hermann Ney 2007a. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. *SSST, NAACL-HLT-2007 AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1-8, Rochester, NY.

Yuqi Zhang, Richard Zens, and Hermann Ney 2007b. Improved chunk-level reordering for statistical machine translation. *IWSLT 2007: International Workshop on Spoken Language Translation*, pages 21-28, Trento, Italy.