Generalized Reordering Rules for Improved SMT

Fei Huang
IBM T. J. Watson Research Center
huangfe@us.ibm.com

Cezar Pendus
IBM T. J. Watson Research Center
cpendus@us.ibm.com

Abstract

We present a simple yet effective approach to syntactic reordering for Statistical Machine Translation (SMT). Instead of solely relying on the top-1 best-matching rule for source sentence preordering, we generalize fully lexicalized rules into partially lexicalized and unlexicalized rules to broaden the rule coverage. Furthermore, , we consider multiple permutations of all the matching rules, and select the final reordering path based on the weighed sum of reordering probabilities of these rules. Our experiments in English-Chinese and English-Japanese translations demonstrate the effectiveness of the proposed approach: we observe consistent and significant improvement in translation quality across multiple test sets in both language pairs judged by both humans and automatic metric.

1 Introduction

Languages are structured data. The proper handling of linguistic structures (such as word order) has been one of the most important yet most challenging tasks in statistical machine translation (SMT). It is important because it has significant impact on human judgment of Machine Translation (MT) quality: an MT output without structure is just like a bag of words. It is also very challenging due to the lack of effective methods to model the structural difference between source and target languages.

A lot of research has been conducted in this area. Approaches include distance-based penalty function (Koehn et. al. 2003) and lexicalized distortion models such as (Tillman 2004), (Al-Onaizan and Papineni 2006). Because these models are relatively easy to compute, they are widely used in phrase-based SMT systems. Hierarchical phrase-based system (Hiero, Chiang, 2005) utilizes long range reordering information without syntax. Other models use more syntactic information (string-to-tree, tree-to-string, tree-to-tree, string-to-dependency etc.) to capture the structural difference between language pairs, including (Yamada and Knight, 2001), (Zollmann and Venugopal, 2006), (Liu et. al. 2006), and (Shen et. al. 2008). These models demonstrate better handling of sentence structures, while the computation is more expensive compared with the distortion-based models.

In the middle of the spectrum, (Xia and McCord 2004), (Collins et. al 2005), (Wang et. al. 2007), and (Visweswariah et. al. 2010) combined the benefits of the above two strategies: their approaches reorder an input sentence based on a set of reordering rules defined over the source sentence's syntax parse tree. As a result, the re-ordered source sentence resembles the word order of its target translation. The reordering rules are either hand-crafted or automatically learned from the training data (source parse trees and bitext word alignments). These rules can be unlexicalized (only including the constituent labels) or fully lexicalized (including both the constituent labels and their head words). The unlexicalized reordering rules are more general and can be applied broadly, but sometimes they are not discriminative enough. In the following English-Chinese reordering rules,

0.44 *NP PP* → 0 1
0.56 *NP PP* → 1 0

the NP and PP nodes are reordered with close to random probabilities. When the constituents are attached with their headwords, the reordering probability is much higher than that of the unlexicalized rules.

0.20 *NP:testimony PP:by* --> 0 1
0.80 *NP:testimony PP:by* --> 1 0

Unfortunately, the application of lexicalized reordering rules is constrained by data sparseness: it is unlikely to train the *NP*:<*noun*>

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 387–392,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

PP:<*prep*> reordering rules for every noun-preposition combination. Even for the learnt lexicalized rules, their counts are also relatively small, thus the reordering probabilities may not be estimated reliably, which could lead to incorrect reordering decisions.

To alleviate this problem, we generalize fully lexicalized rules into partially lexicalized rules, which are further generalized into unlexicalized rules. Such generalization allows partial match when the fully lexicalized rules can not be found, thus achieving broader rule coverage.

Given a node of a source parse tree, we find all the matching rules and consider all their possible reorder permutations. Each permutation has a reordering score, which is the weighted sum of reordering probabilities of all the matching rules. We reorder the child nodes based on the permutation with the highest reordering score. Finally we translate the reordered sentence in a phrase-based SMT system. Our experiments in English to Chinese (**EnZh**) and English to Japanese (**EnJa**) translation demonstrate the effectiveness of the proposed approach: we observe consistent improvements across multiple test sets in multiple language pairs and significant gain in human judgment of the MT quality.

This paper is organized as follows: in section 2 we briefly introduce the syntax-based reordering technique. In section 3, we describe our approach. In section 4, we show the experiment results, which is followed by conclusion in section 5.

2 Baseline Syntax-based Reordering

In the general syntax-based reordering, reordering is achieved by permuting the children of any interior node in the source parse tree. Although there are cases where reordering is needed across multiple constituents, this still is a simple and effective technique.

Formally, the reordering rule is a triple {*p, lhs, rhs*}, where *p* is the reordering probability, *lhs* is the left hand side of the rule, i.e., the constituent label sequence of a parse tree node, and *rhs* is the reordering permutation derived either from hand-crafted rules as in (Collins et. al 2005) and (Wang et. al. 2007), or from training data as in (Visweswariah et. al. 2010).

The training data includes bilingual sentence pairs with word alignments, as well as the source sentences' parse trees. The children's relative order of each node is decided according to their average alignment position in the target sentence. Such relative order is a permutation of the integer sequence [0, 1, … *N*-1], where *N* is the number of children of the given parse node. The counts of each permutation of each parse label sequence will be collected from the training data and converted to probabilities as shown in the examples in Section 1. Finally, only the permutation with the highest probability is selected to reorder the matching parse node. The SMT system is re-trained on reordered training data to translate reordered input sentences.

Following the above approach, only the reordering rule [0.56 *NP PP* → 1 0] is kept in the above example. In other words, all the *NP PP* phrases will be reordered, even though the reordering is only slightly preferred in all the training data.

3 Generalized Syntactic Reordering

As shown in the previous examples, reordering depends not only on the constituents' parse labels, but also on the headwords of the constituents. Such fully lexicalized rules suffer from data sparseness: there is either no matching lexicalized rule for a given parse node or the matching rule's reordering probability is unreliable. We address the above issues with rule generalization, then consider all the permutations from multi-level rule matching.

3.1 Rule Generalization

Lexicalized rules are applied only when both the constituent labels and headwords match. When only the labels match, these reordering rules are not used. To increase the rule coverage, we generalize the fully lexicalized rules into partially lexicalized and unlexicalized rules.

We notice that many lexicalized rules share similar reordering permutations, thus it is possible to merge them to form a partially lexicalized rule, where lexicalization only appears at selected constituent's headword. Although it is possible to have multiple lexicalizations in a partially lexicalized rule (which will exponentially increase the total number of rules), we observe that most of the time reordering is triggered by a single constituent. Therefore we keep one lexicalization in the partially lexicalized rules. For example, the following lexicalized rule:

*VB:**appeal** PP-MNR:**by** PP-DIR:**to*** --> 1 2 0 will be converted into the following 3 partially lexicalized rules:

*VB:**appeal** PP-MNR PP-DIR* --> 1 2 0
*VB PP-MNR:**by** PP-DIR* --> 1 2 0
*VB PP-MNR PP-DIR:**to*** --> 1 2 0

The count of each rule will be the *sum* of the fully lexicalized rules which can derive the given partially lexicalized rule. In the above preordering rules, "MNR" and "DIR" are functional labels, indicating the semantic labels ("manner", "direction") of the parse node.

We could go even further, converting the partially lexicalized rules into unlexicalized rules. This is similar to the baseline syntax reordering model, although we will keep all their possible permutations and counts for rule matching, as shown below.

5 *VB PP-MNR PP-DIR* --> 2 0 1
22 *VB PP-MNR PP-DIR* --> 2 1 0
21 *VB PP-MNR PP-DIR* --> 0 1 2
41 *VB PP-MNR PP-DIR* --> 1 2 0
35 *VB PP-MNR PP-DIR* --> 1 0 2

Note that to reduce the noise from paring and word alignment errors, we only keep the reordering rules that appear at least 5 times. Then we convert the counts into probabilities:

$$p_i(rhs \mid lhs_i) = \frac{C_i(rhs, lhs_i)}{\sum C_i(*, lhs_i)}$$

where $i \in \{f, p, u\}$ represents the fully, partially and un-lexicalized rules, and $C_i(rhs, lhs_i)$ is the count of rule (*lhs*$_i$ → *rhs*) in type *i* rules.

When we convert the most specific *fully lexicalized* rules to the more general *partially lexicalized* rules and then to the most general *unlexicalized* rules, we increase the rule coverage while keep their discriminative power at different levels as much as possible.

3.2 Multiple Permutation Multi-level Rule Matching

When applying the three types of reordering rules to reorder a parse tree node, we find all the matching rules and consider all possible permutations. As multiple levels of rules can lead to the same permutation with different probabilities, we take the weighted sum of probabilities from all matching rules (with the same *rhs*). Therefore, the permutation decision is not based on any particular rule, but the combination of all the rules matching different levels of context. As opposed to the general syntax-based reordering approaches, this strategy achieves a desired balance between broad rule coverage and specific rule match: when a fully lexicalized rule matches, it has strong influence on the permutation decision given the richer context. If such specific rule is unavailable or has low probability, more general (partial and unlexicalized) rules will have higher weights. For each permutation we compute the weighted reordering probability, then select the permutation that has the highest score.

Formally, given a parse tree node *T*, let lhs_f be the label:head_word sequence of the fully lexicalized rules matching *T*. Similarly, lhs_p and lhs_u are the sequences of the matching partially lexicalized and unlexicalized rules, respectively, and let *rhs* be their possible permutations. The top-score permutation is computed as:

$$rhs^* = \arg\max_{rhs} \sum_{i \in \{f,p,u\}} w_i p_i(rhs \mid lhs_i)$$

where w_i's are the weights of different kind of rules and p_i is reordering probability of each rule. The weights are chosen empirically based on the performance on a held-out tuning set. In our experiments, w_f=1.0, w_p=0.5, and w_u=0.2, where higher weights are assigned to more specific rules.

For each parse tree node, we identify the top permutation choice and reorder its children accordingly. The source parse tree is traversed breadth-first.

4 Experiments

We applied the generalized syntax-based reordering on both English-Chinese (EnZh) and English-Japanese (EnJa) translations. Our English parser is IBM's maximum entropy constituent parser (Ratnaparkhi 1999) trained on Penn Treebank. Experiments in (Visweswariah et. al. 2010) indicated that minimal difference was observed using Berkeley's parser or IBM's parser for reordering.

Our EnZh training data consists of 20 million sentence pairs (~250M words), half of which are from LDC released bilingual corpora and the other half are from technical domains (e.g., software manual). We first trained automatic word alignments (HMM alignments in both directions and a MaxEnt alignment (Ittycheriah and Roukos, 2005)), then parsed the English sentences with the IBM parser. We extracted different reordering rules from the word alignments and the English parse trees. After frequency-based pruning, we obtained 12M lexicalized rules, 13M partially lexicalized rules and 600K unlexicalized rules. Using these rules, we applied preordering on the English sentences and then built an SMT system with the reordered training data. Our decoder is a phrase-based decoder (Tillman 2006), where various features are combined within the log-linear framework. These features include source-to-target phrase translation score based on relative frequency, source-to-target and target-to-source word-to-word translation scores, a 5-gram language model score, distortion model scores and word count.

	Tech1	**Tech2**	**MT08**
# of sentences	582	600	1859
PBMT	33.08	31.35	36.81
UnLex	33.37	31.38	36.39
FullLex	34.12	31.62	37.14
PartLex	34.13	32.58	37.60
MPML	**34.34**	**32.64**	**38.02**

Table 1: MT experiment comparison using different syntax-based reordering techniques on English-Chinese test sets.

We selected one tuning set from software manual domain (**Tech1**), and used PRO tuning (Hopkins and May 2011) to select decoder feature weights. Our test sets include one from the online technical support domain (**Tech2**) and one from the news domain: the NIST MT08 English-Chinese evaluation test data. The translation quality is measured by BLEU score (Papineni et. al., 2001). Table 1 shows the BLEU score of the baseline phrase-based system (**PBMT**) that uses lexicalized reordering at decoding time rather than *pre*ordering. Next, Table 1 shows the translation results with several preordered systems that use unlexicalized (**UnLex),** fully lexicalized (**FullLex**) and partially lexicalized (**PartLex**) rules, respectively. The lexicalized reordering model is still applicable for preordered systems so that some preordering errors can be recovered at run time.

First we observed that the **UnLex** preordering model on average does not improve over the typical phrase-based MT baseline due to its limited discriminative power. When the preordering decision is conditioned on the head word, the **FullLex** model shows some gains (~0.3 pt) thanks to the richer matching context, while the **PartLex** model improves further over the **FullLex** model because of its broader coverage. Combining all three with multi-permutation, multi-level rule matching (**MPML**) brings the most gains, with consistent (~1.3 Bleu points) improvement over the baseline system on all the test sets. Note that the Bleu scores on the news domain (**MT08**) are higher than those on the tech domain. This is because the Tech1 and Tech2 have one reference translation while MT08 has 4 reference translations.

In addition to the automatic MT evaluation, we also used human judgment of quality of the MT translation on a set of randomly selected 125 sentences from the baseline and improved reordering systems. The human judgment score is 2.82 for the **UnLex** system output, and 3.04 for the improved **MPML** reordering output. The 0.2 point improvement on the 0-5 scale is considered significant.

	Tech1	**Tech2**	**News**
# of sentences	1000	600	600
PBMT	56.45	35.45	21.70
UnLex	59.22	38.36	23.08
FullLex	57.55	36.56	22.23
PartLex	59.80	38.47	23.13
MPML	**59.94**	**38.62**	**23.31**

Table 2: MT experiment comparison using generalized syntax-based reordering techniques on English-Japanese test sets.

We also apply the same generalized reordering technique on English-Japanese (**EnJa**) translation. As there is very limited publicly available English-Japanese parallel data, most our training data (20M sentence pairs) is from the in-house software manual domain. We use the same English parser and phrase-based decoder as in EnZh experiment. Table 2 shows the translation results on technical and news domain test sets. All the test sets have single reference translation.

First, we observe that the improvement from preordering is larger than that in EnZh MT (1.6-3 pts vs. 1 pt). This is because the word order difference between English and Japanese is larger than that between English and Chinese (Japanese is a SOV language while both English and Chinese are SVO languages). Without preordering, correct word orders are difficult to obtain given the typical skip-window beam search in the **PBMT**. Also, as in EnZh, the **PartLex** model outperforms the **UnLex** model, both of which being significantly better than the **FullLex** model due to the limited rule coverage in the later model: only 50% preordering rules are applied in the **FullLex** model. **Tech1** test set is a very close match to the training data thus its BLEU score is much higher.

5 Conclusion and Future Work

To summarize, we made the following improvements:

1. We generalized fully lexicalized reordering rules to partially lexicalized and unlexicalized rules for broader rule coverage and reduced data sparseness.
2. We allowed multiple permutation, multi-level rule matching to select the best reordering path.

Experiment results show consistent and significant improvements on multiple English-Chinese and English-Japanese test sets judged by both automatic and human judgments.

In future work we would like to explore new methods to prune the phrase table without degrading MT performance and to make rule extraction and reordering more robust to parsing errors.

Acknowledgement

The authors appreciate helpful comments from anonymous reviewers as well as fruitful discussions with Karthik Visweswariah and Salim Roukos.

References

Yaser Al-Onaizan , Kishore Papineni, Distortion models for statistical machine translation, Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, p.529-536, July 17-18, 2006, Sydney, Australia

David Chiang, A hierarchical phrase-based model for statistical machine translation, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, p.263-270, June 25-30, 2005, Ann Arbor, Michigan

Michael Collins , Philipp Koehn , Ivona Kucerov, Clause restructuring for statistical machine translation, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, p.531-540, June 25-30, 2005, Ann Arbor, Michigan

Mark Hopkins, Jonathan May, Tuning as ranking, In Proceedings of the Conference on Empirical Methods in Natural Language Processing 2011, pp. 1352-1362. Association for Computational Linguistics.

Abraham Ittycheriah , Salim Roukos, A maximum entropy word aligner for Arabic-English machine translation, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, p.89-96, October 06-08, 2005, Vancouver, British Columbia, Canada

Philipp Koehn , Franz Josef Och , Daniel Marcu, Statistical phrase-based translation, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, p.48-54, May 27-June 01, 2003, Edmonton, Canada

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In Proceedings of COLING/ACL 2006, pages 609-616, Sydney, Australia, July.

Libin Shen, Jinxi Xu and Ralph Weischedel 2008. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. in Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL). Columbus, OH, USA, June 15 - 20, 2008.

Christoph Tillmann, A unigram orientation model for statistical machine translation, Proceedings of HLT-NAACL 2004: Short Papers, p.101-104, May 02-07, 2004, Boston, Massachusetts

Christoph Tillmann. 2006. Efficient Dynamic Programming Search Algorithms for Phrase-based SMT. In Proc. of the Workshop CHPSLP at HLT'06.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In Proceedings of EMNLP-CoNLL.

Karthik Visweswariah , Jiri Navratil , Jeffrey Sorensen , Vijil Chenthamarakshan , Nanda Kambhatla, Syntax based reordering with automatically derived rules for improved statistical machine translation, Proceedings of the 23rd International Conference on Computational Linguistics, p.1119-1127, August 23-27, 2010, Beijing, China

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. Machine Learning, 34(1-3).

Fei Xia , Michael McCord, Improving a statistical MT system with automatically learned rewrite patterns, Proceedings of the 20th international conference on Computational Linguistics, p.508-es, August 23-27, 2004, Geneva, Switzerland

Kenji Yamada , Kevin Knight, A syntax-based statistical translation model, Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, p.523-530, July 06-11, 2001, Toulouse, France

Andreas Zollmann , Ashish Venugopal, Syntax augmented machine translation via chart parsing, Proceedings of the Workshop on Statistical Machine Translation, June 08-09, 2006, New York City, New YorkAlfred. V. Aho and Jeffrey D. Ullman. 1972. The Theory of Parsing, Translation and Compiling, volume 1. Prentice-Hall, Englewood Cliffs, NJ.