A Framework for Effectively Integrating Hard and Soft Syntactic Rules into Phrase Based Translation*

Jiajun Zhang and Chengqing Zong

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{jjzhang, cqzong}@nlpr.ia.ac.cn

Abstract. In adding syntactic knowledge into phrase-based translation, using hard or soft syntactic rules to reorder the source-language aiming to closely approximate the target-language word order has been successful in improving translation quality. However, it suffers from propagating the pre-reordering errors to the later translation step (decoding). In this paper, we propose a novel framework to integrate hard and soft syntactic rules into phrase-based translation more effectively. For a source sentence to be translated, hard or soft syntactic rules are first acquired from the source parse tree prior to translation, and then instead of reordering the source sentence directly, the rules are used as a strong feature integrated into our elaborately designed model to help phrase reordering in the decoding stage. The experiments on NIST Chinese-to-English translation show that our approach, whether incorporating hard or soft rules, significantly outperforms the previous methods.

Keywords: hard syntactic rules, soft syntactic rules, effective integration, phrase-based translation

1 Introduction

Adding syntax into phrase-based translation has become a hot research topic. Many works, such as (Collins et al., 2005; Wang et al., 2007; Cherry 2008; Marton and Resnik, 2008; and Badr, 2009), have investigated how to use the linguistic information in phrase-based SMT and empirically proved that syntactic knowledge is very helpful to improve translation performance especially in phrase reordering. For example, in Chinese-to-English translation, the Chinese phrase **PP-VP** is translated into English **VP-PP** in most cases. Thus, if a special rule is designed to deal with the case of this kind, the translation result will be better.

The popular way of integrating the linguistic information into phrase reordering is to reorder the source sentences with syntactic reordering rules so as to make the input much closer to the target language in word order. (Collins et al., 2005; Wang et al, 2007 and Badr et al., 2009) used **hard syntactic rules** (namely manually created) obtained from source parse trees to directly reorder the input sentences. (Li et al., 2007) employed **soft syntactic rules** (namely probabilistic) to get an *n*-best reordered sentence list for decoding. The former method depends much on the author's professional knowledge in linguistics and the performance in parsing technology. The latter approach is more robust to the errors in parsing stage but increases the burden of decoding as it has to translate an *n*-best sentences, and furthermore, it might still produce pre-reordering errors prior to translation because the *n*-best list includes only part of but not all of the reordering hypotheses. It should be noted that both the two methods are implemented directly in parse trees, and it is pointed out in previous work (Habash, 2007) that

* We would like to thank Yu Zhou for her suggestions to revise the earlier draft and thank anonymous reviewers for their helpful comments. The research work has been partially funded by the Natural Science Foundation of China under grant No.60736014, 60723005 and 90820303, the National Key Technology R&D Program under grant No. 2006BAH03B02, the Hi-Tech Research and Development Program (863 Program) of China under grant No. 2006AA010108-4, and also supported by the China-Singapore Institute of Digital Media as well.

syntactic reordering does not improve translation if the parse quality is not good enough. Therefore, it becomes a challenge that how to use the hard and soft syntactic rules properly and adequately even though the parse quality is not very good (taking Chinese parsers as an example).

It is natural that many researchers apply syntactic rules rather than distortion or lexical features to improve phrase reordering because the syntactic knowledge is more reliable. However, due to the parsing errors and the discrepancy between translation units and syntactic rules, reordering the source sentences prior to translation could cause many errors which might not be made up for in later translation steps. For example, the Chinese parser would mistakenly parse the Chinese noun phrase "NP(以$_{\text{NN}}$(israeli) 巴$_{\text{NN}}$(palestinian) 和平$_{\text{NN}}$(peace))" into a prepositional phrase "PP(以$_{\text{P}}$(with) 巴$_{\text{NN}}$(palestinian) 和平$_{\text{NN}}$(peace))", and the pre-reordering hard rules[1] will wrongly move this fake prepositional phrase after its right sibling verb phrase if any, so the translation would be wrong.

Our motivation is based on the above analysis. Instead of using these syntactic rules to reorder the source sentences arbitrarily, we use them as a strong feature integrated into our finely designed model to guide phrase reordering in decoding stage and meanwhile create an extra feature to reward the syntactic reordering during decoding. Thus, we not only utilize the good syntactic rules adequately, but also make up for the bad syntactic rules with other important features such as phrase translation probability and target language model. Moreover, it does not increase the time complexity of decoding.

In the model construction, we still employ the log-linear model to combine translation model, target language model and reordering model. The difference lies in two aspects: on the one hand, we divide the reordering model into syntactic reordering model and non-syntactic one in order to easily integrate syntactic rules. On the other hand, we add an extra feature to reward syntactic reordering so as to emphasize the importance of syntactic rules. For a source sentence to be translated, our framework of translation can be illustrated in Figure 1(b).

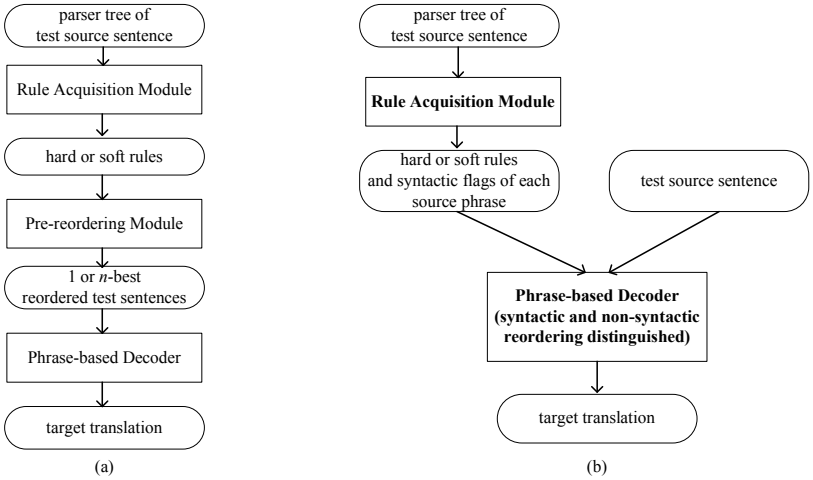

Figure 1: (a) shows the translation flowchart of previous pre-reordering methods. (b) illustrates our translation framework of incorporating hard or soft rules into the decoding stage. We will detail respectively the two key parts which are in **boldface** in Section 3 and Section 4.

To verify the competitiveness of our approach, we have developed two systems: one uses this approach to integrate hard syntactic rules, and the other employs the approach to incorporate soft syntactic rules. The two systems will be compared with those using the previous methods.

We introduce the related work in Section 2. Section 3 describes the acquisition and representation of syntactic rules. Section 4 details the integration algorithm of syntactic rules

[1] The hard rules will be detailed in Section 3.1

into the decoding module. In Section 5, we discuss the experiments and analysis. Section 6 concludes the paper.

2 Related Work

In recent years, it has been widely studied on how to incorporate the syntactic information of source language to improve phrase reordering.

Collins et al. (2005) described six types of transforming rules to reorder the German clauses in German-to-English translation. Wang et al. (2007) analyzed the systematic difference between Chinese and English and proposed specific pre-reordering rules for three categories of Chinese phrase: verb phrases, noun phrases, and localizer phrases. Badr (2009) addressed two syntactic constructs (Subject-Verb structure and noun phrase structure) and exploited well-defined pre-reordering rules for English-to-Arabic translation. However, all the rules in the above three methods are hard ones (manually built) and sometimes cause many pre-reordering errors. In order to improve the robustness, Li et al. (2007) used the weighted reordered *n*-best source sentences as input for the decoder. They utilized the soft rules based on source parse trees in Chinese-to-English translation to determine whether the children of a node should be reordered or not, and finally to obtain a reordered *n*-best list. However, all these methods are separated from decoder and reorder the source sentences arbitrarily prior to translation. Once a pre-reordering error happens, it is very difficult to make up for the mistake in later translation steps. In our approach, we just retain the syntactic rules rather than use them to reorder the source sentences directly. During decoding, the syntactic rules will serve as a strong feature to guide and enhance the phrase reordering.

Zhang et al., (2007) only allowed reordering between syntactic phrases and enforced the non-syntactic phrases translated in order. Xiong et al. (2008) proposed a linguistically annotated BTG for SMT. The method used some heuristic rules to linguistically annotate every source phrase with the source-side parse tree in decoding and built a linguistical reordering model. The two approaches both acquired and applied the syntactic rules in the decoding stage but meanwhile increased the decoding time to a large extent. Our work differs from theirs in three ways. First, when translating a test sentence, we obtain the corresponding syntactic rules prior to translation rather than in decoding stage and thus alleviate the decoding complexity. Second, we distinguish syntactic reordering from non-syntactic reordering because we believe they play different roles in translation. We think this idea is not considered in previous works. Third, we add an extra feature to reward the syntactic reordering.

3 Acquisition and Representation of Syntactic Rules

In this paper, we use Chinese-to-English translation as a case study. However, our approach is also suited for other language translations only if syntactic rules of the test sentence are provided. Whether incorporate hard syntactic rules or soft syntactic rules, obtaining these rules is our first task.

3.1 Hard Rule Acquisition

The hard syntactic rules which are handcrafts and do not need to be trained should reflect the true structural difference between the two languages Chinese and English. (Wang et al., 2007) described three kinds of hard rules for Chinese-to-English which we think are reasonable. Here, we revisit and conclude these specific rules.

- **Verb Phrases** If there is a node in Chinese parse tree labeled as VP[2], we have three rules to reorder its children. (1) VP(PP◇VP)→VP(◇VP PP)[3] and VP(LCP◇VP)→VP(◇VP LCP) indicate that PP or LCP in a parent VP needs to be repositioned after the sibling VP. (2)

[2] All the phrase labels we use are borrowed from Penn Chinese Treebank phrase tags

[3] The notation ◇ is a placeholder which indicates other syntactic nodes between PP and VP

VP(NP(NT) ◇VP)→VP(◇VP NP(NT)) means a preverbal NP should be moved after the sibling VP if there is at least one NT in the NP subtree. (3) VP(QP◇VP)→VP(◇VP QP) states QP in a parent VP will be repositioned after the sibling VP.

- **Noun Phrases** When we find a NP node in Chinese parse tree, four rules are considered. (1) NP(DNP(PP|LCP)◇NP)→NP(◇NP DNP(PP|LCP)) indicates that DNP is repositioned after the last sibling NP if a parent NP has a child DNP which in turn has a child PP or LCP. (2) NP(DNP(!PN) ◇NP)→NP(◇NP DNP(!PN)) denotes that if a parent NP has a child DNP which in turn has a child NP that is not a PN, then the DNP should be moved after the last sibling NP. (3) NP(CP◇NP)→NP(◇NP CP) means the child CP will be repositioned after its sibling NP. (4) CP(IP DEC)→CP(DEC IP) says that if CP in rule (3) is formed by "IP+DEC", we have to exchange these two nodes.
- **Localizers** We have one rule for the node LCP: LCP(◇LC)→LCP(LC◇) denoting the child LC node will be moved before its left sibling under a parent LCP node.

Given the parse tree of a test source sentence, we can extract all the hard rules belonging to the above ones. Taking a Chinese sub-sentence and its parse tree below as an example, there exists a hard rule VP(PP◇VP)→VP(◇VP PP) in which PP is (到$_2$ 新$_3$ 的$_4$ 办公$_5$ 大楼$_6$) and VP is (是$_8$ 一个$_9$挑战$_{10}$). Note that if we apply pre-reordering method to reorder the input sentence and move PP after VP, we may get a wrong translation "relocation will be a challenge to a new office building" because the syntactic tree is parsed with error.

Chinese: 迁移 到 新 的 办公 大楼 将 是 一个 挑战

Chinese pinyin: qiānyí dào xīn de bàngōng dàlóu jiāng shì yīgè tiǎozhàn

English: relocation to a new office building will be a challenge

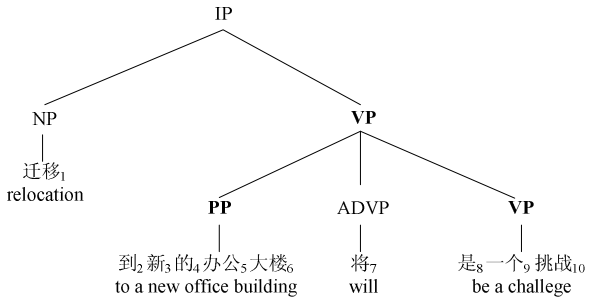

Figure 2: The simplified Chinese parse tree of the example sentence and the leaves are Chinese words with their index and corresponding English translation.

3.2 Soft Rule Acquisition

About the soft rules, we use a similar way with (Li et al., 2007) to extract them and predict their probabilities. Li et al., (2007) only concern the nodes with two or three children and predict a probability for each permutation of the children. We turn to another strategy. For the nodes with two children, we just design a rule to determine whether they should be reordered. For the nodes with more than two children, we first search the central node (VP or NP), if it exists, we design a rule to decide whether any preceding modifier node should be repositioned after the central node. The second rule is based on the phenomenon that the modifiers before VP or NP in Chinese usually appear after VP or NP in English. The two rules can be formalized as following:

$$\mathrm{P}: \mathrm{N}^{\mathrm{l}} \diamond \mathrm{N}^{\mathrm{r}} \Rightarrow \begin{cases} \mathrm{N}^{\mathrm{l}} \diamond \mathrm{N}^{\mathrm{r}} & \text{straight} \\ \diamond \mathrm{N}^{\mathrm{r}} \mathrm{N}^{\mathrm{l}} & \text{inverted} \end{cases} \quad (1)$$

Where ◇ is NULL if the parent node P has two children (left node N^{l} and right node N^{r}), or is other nodes between modifier node N^{l} and central node N^{r} if P has more than two children.

For the two kinds of soft rules, we adopt a maximum entropy (ME) model to predict their probabilities. When extracting training examples, we use the Chinese parse tree and the word

alignment between Chinese and English as input. If the English sides aligned to the two Chinese nodes we handled are not crossing, a training example can be extracted. The rich features we employ for ME training and predicting include leftmost/rightmost word of N^l and N^r, the part-of-speech of these words, the word immediately before/after the leftmost/rightmost word of N^l/ N^r plus the combining phrase tags of N^l, N^r and their parent. Taking the rule used in section 3.1 as an example, namely N^l = PP (到$_2$ 新$_3$ 的$_4$ 办公$_5$ 大楼$_6$) and N^r = VP(是$_8$一个$_9$挑战$_{10}$). The specific features about this rule are listed in Table 1.

Given the parse tree of a test source sentence, we first extract all the soft rules and then predict their probabilities with the trained ME model. For the pre-reordering method, these soft rules are employed to produce an *n*-best reordered source sentences as the input of the decoder. In our approach, we apply these rules to guide phrase reordering in the decoding stage.

Table 1: The specific features for a rule, "l/r" denotes leftmost/rightmost, "w" means word, "p" indicates part-of-speech, and "b/a" means before/after.

lw of N^l	rw of N^l	lp of N^l	rp of N^l	lw of N^r	rw of N^r	lp of N^r	rp of N^r	bw of N^l	aw of N^r	tag of rule
到	大楼	P	NN	是	挑战	VV	NN	迁移	NULL	PP-VP-VP

3.3 Rule Representation

Let us first have a review of the forms of the hard and soft syntactic rules. The hard syntactic rules have the form like VP(PP◇VP)→VP(◇VP PP), CP(IP DEC)→CP(DEC IP) and LCP(◇LC)→LCP(LC◇). It should be noted that ◇ in the last rule cannot be NULL and we regard it as a special node. Therefore, all the hard rules are binary relations between two nodes. It is the same relation in the soft syntactic rules which use the forms $\langle N^l \diamond N^r \rightarrow N^l \diamond N^r, P(s)\rangle$and $\langle N^l \diamond N^r \rightarrow \diamond N^r N^l, P(i)\rangle$ where P(s) and P(i) denotes probabilities of straight and inverted respectively. It is obvious and easy to change the hard rule into an equivalent probabilistic format. For example, VP(PP◇VP)→VP(◇VP PP) is equivalent to $\langle PP \diamond VP \rightarrow \diamond VP\ PP, 1.0\rangle$. Thus, we can see that the hard rule is a special case of soft rule, and the only difference lies in that the hard rule only has the inverted format.

For the sake of convenience, hereafter, we only consider the generalized rule formats $\langle N^l \diamond N^r \rightarrow N^l \diamond N^r, P(s)\rangle$ and $\langle N^l \diamond N^r \rightarrow \diamond N^r N^l, P(i)\rangle$. Since $P(s) + P(i) = 1.0$, we can use only one format to denote these two ones. It is $\langle N^l, N^r, P(i)\rangle$ which means the left node N^l will be repositioned after the right node N^r with the probability P(i). $P(i) = 1.0$ if it is a hard rule, otherwise P(i) is predicted by ME model. As the unit of phrase-based translation is a source phrase but not a parse tree node, we have to make a conversion from tree nodes to source phrases in order to incorporate the syntactic rules. Since each tree node can be projected to be a span on the source sentence, we can just use spans to denote the tree nodes. Finally, any syntactic rule can be represented as a triple $\langle span(N^l), span(N^r), P(i)\rangle$.

4 Integrating Syntactic Rules

We integrate the syntactic rules into a phrase-based SMT to help the decoder performs more linguistically. In this paper, we choose the decoder with Bracket Transduction Grammar (BTG) style model (Wu, 1997; Xiong et al., 2006) as our baseline.

4.1 BTG-based Model

The BTG-based translation can be viewed as a monolingual parsing process, in which only lexical rules $A \rightarrow (x, y)$ and two binary merging rules $A \rightarrow [A^l, A^r]$ and $A \rightarrow \langle A^l, A^r\rangle$ are allowed.

During decoding, the source sentence is first divided into phrase sequence, then the lexical rule $A \rightarrow (x, y)$ translates the source phrase x into target phrase y and forms a block A. The

straight rule $A \rightarrow [A^l, A^r]$ and the inverted rule $A \rightarrow \langle A^l, A^r \rangle$ merge the two neighboring blocks into a bigger one until the whole source sentence is covered. It is natural to adopt a CKY-style algorithm for this decoding process. The straight rule requires the order of two blocks in source and target language consistent, while the inverted rule swaps the target parts of the two blocks. Score of the lexical rule is computed as follows:

$$Pr^l(A) = p(y|x)^{\lambda_1} \cdot p(x|y)^{\lambda_2} \cdot p_{lex}(y|x)^{\lambda_3} \cdot p_{lex}(x|y)^{\lambda_4} \cdot exp(l)^{\lambda_5} \cdot exp(|y|)^{\lambda_6} \cdot P_{LM}^{\lambda_7}(y) \quad (2)$$

Where the first two factors are bidirectional phrase translation probabilities, $p_{lex}(y|x)$ and $p_{lex}(x|y)$ denote bidirectional word translation probabilities, $exp(l)$ and $exp(|y|)$ denote phrase number penalty and the target length penalty respectively and $P_{LM}(y)$ is the probability of target language model. The λs are their corresponding feature weights.

We compute the score of merging rules as:

$$Pr^m(A) = \Omega^{\lambda_8} \cdot P_{LM}^{\lambda_7}(y) \quad (3)$$

In which Ω is the reordering score and λ_8 is its weight. Similar to (Xiong et al., 2006), the reordering score is calculated by the ME model with only lexical boundary words (leftmost and rightmost) of phrases as features.

4.2 Model Adaptation for Syntactic Rules

We first give the definition of syntactic phrase and non-syntactic phrase in this section. The phrase that exactly covers a sub-tree of source parse tree is defined as a **syntactic phrase**. The phrase covering continuous child nodes of a tree node is also considered as a syntactic phrase. Other phrases are regarded as **non-syntactic phrases**.

According to the definition, the syntactic rules are all about reordering between syntactic phrases. Our basic idea of integrating the syntactic rule is to use its probability as the phrase reordering probability if the merging phrases match the syntactic rule. Therefore, the syntactic rules only influence syntactic phrase reordering. The features the baseline reordering model use are just lexical boundary words, while our syntactic rules embedded much more linguistical features. Thus, we believe the syntactic phrase reordering plays a more important role than non-syntactic one and they should be distinguished from each other. The new score of merging rules will be computed as follows:

$$Pr^m(A) = {\Omega_N}^{\lambda_8 \cdot I_N(A)} \cdot {\Omega_S}^{\lambda_9 \cdot I_S(A)} \cdot P_{LM}^{\lambda_7}(y) \quad (4)$$

Where Ω_S and Ω_N are syntactic and non-syntactic reordering score respectively. $I_S(A)$ and $I_N(A)$ are indicator functions which indicate that Ω_S is used when A is merging two syntactic phrases, otherwise Ω_N is employed.

To emphasize the importance of syntactic phrase reordering, we further create another feature to enhance syntactic reordering (because weights tuning cannot promise the weight of syntactic reordering model bigger and more importance than that of non-syntactic reordering model). The final score of merging rules are calculated as follows:

$$Pr^m(A) = {\Omega_N}^{\lambda_8 \cdot I_N(A)} \cdot {\Omega_S}^{\lambda_9 \cdot I_S(A)} \cdot R_S^{\lambda_{10}} \cdot P_{LM}^{\lambda_7}(y) \quad (5)$$

In which R_S is a binary feature in order to reward syntactic reordering and it equals to 1 if Ω_S is active. All the ten feature weights $\lambda_1 \sim \lambda_{10}$ in our new model are tuned with MERT (Och, 2003).

4.3 Algorithm of Integrating Syntactic Rules

After knowing the translation model and the decoding algorithm we have used, the most important thing we care about is how to integrate the syntactic rules during decoding.

The ultimate format of syntactic rule we adopt is designed as $\langle \mathrm{span}(\mathrm{N}^{\mathrm{l}}),\ \mathrm{span}(\mathrm{N}^{\mathrm{r}}), \mathrm{P(i)} \rangle$, and the merging rules used in decoding always handle two continuous phrases, so if $\mathrm{span}(\mathrm{N}^{\mathrm{l}})$ and $\mathrm{span}(\mathrm{N}^{\mathrm{r}})$ are successive, then P(i) will be used to replace the syntactic reordering score Ω_S which is predicted with lexical boundary words in baseline. However, $\mathrm{span}(\mathrm{N}^{\mathrm{l}})$ and $\mathrm{span}(\mathrm{N}^{\mathrm{r}})$

will not be consecutive if there is a non-empty ◇ between the two nodes. A simple strategy is developed to solve this non-continuous problem.

Transformation strategy: We take a soft syntactic rule in Figure 1 as an example to illustrate this detailed strategy. The original rule format is $\langle \mathrm{span}(\mathrm{N}^{\mathrm{l}}), \mathrm{span}(\mathrm{N}^{\mathrm{r}}), \mathrm{P(i)} \rangle$ in which $\mathrm{N}^{\mathrm{l}} = \mathrm{PP}$ (到$_2$ 新$_3$ 的$_4$ 办公$_5$ 大楼$_6$) and $\mathrm{N}^{\mathrm{r}} = \mathrm{VP}$(是$_8$一个$_9$挑战$_{10}$), and so the real rule is $\langle (2,6), (8,10), \mathrm{P(i)} \rangle$ and these two spans are not continuous. Fortunately, it is natural to see the fact that if we reorder the rule $\langle (2,6),\ (8,10), \mathrm{P(i)} \rangle$, the span $(2,10)$ will be $(7,10)$ followed by $(2,6)$ and the result is the same with the inverted case for spans $(2,6)$ and $(7,10)$. Therefore, the rule $\langle (2,6), (8,10), \mathrm{P(i)} \rangle$ is equivalent to $\langle (2,6),\ (7,10), \mathrm{P(i)} \rangle$ in which the spans are consecutive. Thus, for a discontinuous syntactic rule $\langle (\mathrm{i},\mathrm{k}),\ (\mathrm{h},\mathrm{j}), \mathrm{P(i)} \rangle$ where $\mathrm{i} \le \mathrm{k} < h < j$ and $\mathrm{h} \neq \mathrm{k}+1$, we can simply convert it into an equivalent format $\langle (\mathrm{i},\mathrm{k}), (\mathrm{k}+1,\mathrm{j}), \mathrm{P(i)} \rangle$.

Integrating syntactic rules: During decoding, when the CKY algorithm translate the source span(i,j), and at the same time there is a syntactic rule $\langle (\mathrm{i},\mathrm{k}),\ (\mathrm{h},\mathrm{j}), \mathrm{P(i)} \rangle$ matches the span, then we first convert the rule into a continuous one $\langle (\mathrm{i},\mathrm{k}),\ (\mathrm{k}+1,\mathrm{j}), \mathrm{P(i)} \rangle$, and finally $\mathrm{P(i)}$ is utilized as a more reliable score to replace the syntactic reordering score Ω_S predicted with only lexical boundary words as features in baseline.

5 Experiments and Analysis

5.1 Baselines Used

The first baseline is the BTG-based translation system which uses a lexicalized reordering model trained with Maximum Entropy and it is re-implemented according to (Xiong et al., 2006). We denote this baseline as **MEBTG**. We modified the baseline decoder (**MBDecoder**) to incorporate the hard syntactic rules or soft syntactic rules as described as Section 4.2 and 4.3.

To show the competitiveness of our approach, we have to compare our usage of hard syntactic rules with the previous usage in (Wang, et al., 2007), and compare our method of using soft syntactic rules with the previous method in (Li et al., 2007). The classical implementation of the previous usage of syntactic rules is to reorder the source sentences of training, development and test data, then train the translation model with reordered source training data, tune the weights of features with reordered source development data, and at last use a phrase-based system (BTG-based system in this paper) to get the target translation of the reordered test data. The system using hard rules is named **MEBTG+HRP** which means **MEBTG** system with **H**ard **R**ules **P**re-Reordering. Likewise, the system using soft rules is called **MEBTG+SRP** indicating **MEBTG** system with **S**oft **R**ules **P**re-reordering (only 1-best reordered source sentence used for source-side of training data and 10-best for test data).

5.2 Corpora and Experimental Settings

We carried out the experiments on Chinese-to-English translation using NIST05 test set. The development set including 571 Chinese sentences is chosen from the test set of NIST06 and NIST08. The training set consists of 297K parallel sentences which are filtered from LDC.

Word-level alignments were obtained using GIZA++ (Och and Ney, 2000). The target four-gram language model was built with the English part of training data using the SRI Language Modeling Toolkit (Stolcke, 2002). In order to acquire syntactic rules, we parse the Chinese sentence using the Stanford parser (Klein and Manning, 2003) with its default Chinese grammar. We built the maximum entropy model with a MaxEnt Toolkit developed by (Zhang, 2004).

All the models were optimized and tested using the case-sensitive BLEU-4 with "shortest" reference length. Statistical significance in BLEU score difference was measured by using paired bootstrap re-sampling (Koehn, 2004).

5.3 Experimental Results

Before giving the experimental results, some notations of our new systems are first introduced here. The system **IN**corporating the **H**ard **R**ules into the **Modified Baseline Decoder** is named

IN-HR-MBDecoder. Likewise, **IN-SR-MBDecoder** is used to denote the system incorporating the soft rules into modified baseline decoder.

In Table 2, we present our results. Like (Wang et al., 2007) and (Zhang et al., 2007), we find that reordering the source sentences whether with hard rules or with soft rules can both obtain a significant improvement over the baseline MEBTG by 0.58 and 0.60 BLEU respectively. As these two approaches may cause many pre-reordering errors, the gain is not very promising. However, after using our new approach, the system integrating the hard rules into the modified decoder IN-HR-MBDecoder achieves a larger improvement of up to 1.02 BLEU over MEBTG, and also significantly outperforms the system pre-reordering with the hard rules. Furthermore, the system incorporating with the soft rules IN-SR-MBDecoder performs even better. It outperforms MEBTG and MEBTG+SRP both significantly by 1.35 and 0.75. The significant improvements of IN-HR-MBDecoder and IN-SR-MBDecoder indicate that our approach of using syntactic rules as a strong feature to help phrase reordering in the decoding stage is more effective than the previous approach of using them for pre-reordering.

Table 2: Translation results on development set and test set. * or **: significantly better than baseline MEBTG ($p < 0.05$ or $p < 0.01$ respectively). +: significantly better than MEBTG+HRP ($p < 0.05$). ##: significantly better than MEBTG+SRP ($p < 0.01$).

System	Dev	Test
MEBTG	0.2567	0.3296
MEBTG+HRP	0.2635	0.3354*
MEBTG+SRP	0.2652	0.3356*
IN-HR-MBDecoder	0.2671	**0.3398**+
IN-SR-MBDecoder	0.2713	**0.3431**##

Table 3: The effect of new features. "SynNon" means syntactic and non-syntactic reordering model; "SR" denotes soft rules integrated. * or **: significantly better than baseline MEBTG ($\mathbf{p} < 0.05$ or $\mathbf{p} < 0.01$ respectively). @@: significantly better than "SynNon" ($\mathbf{p} < 0.01$).

Features	BLEU-4
SynNon	0.3347*
SynNon+SR	0.3416**@@
SynNon+SR+Reward	0.3431**@@

5.4 Analysis

In this section, we have a detailed analysis about the translation results.

- **Why MEBTG+HRP and MEBTG+SRP perform similar?**

It is interesting that pre-reordering with the hard rules has a similar performance with pre-reordering using the soft rules. We find that because of many Chinese parsing errors, the accuracy of the hard rules is not high, only 62.1% reported in (Wang et al., 2007). So, it causes many pre-reordering errors. Although the system pre-reordering with soft rules does not produce as many errors as MEBTG+HRP does, it may miss some correct reordering instances. Thus, the two systems would have similar translation quality. Two translation instances are illustrated in Figure 3 and Figure 4 to show the situations which hard rules and soft rules may run into.

- **Why IN-SR-MBDecoder outperforms IN-HR-MBDecoder?**

Compared with the systems using syntactic rules for pre-reordering, why our usage of syntactic rules for hard and soft rules could yield a bigger gap (0.33 vs. 0.02)? We know that the two systems are almost the same except that they incorporate different syntactic rules, soft rules versus hard ones. Instead of using them to directly reorder the source sentence, we use them to help phrase reordering in the decoding stage with the same algorithm. Therefore, we believe the difference might lie in the number of rules they have employed. We find that only average 4.18

hard rules are acquired from each test sentence, while 17.08 soft rules in average are obtained. During decoding, the more syntactic information, the better phrase reordering.

- **The effect of new features?**

As described in Section 4.2 and 4.3, our system has three new features: (1) syntactic phrase reordering model and non-syntactic one are employed to replace the baseline reordering model; (2) a binary rewarding feature is used to enhance the syntactic reordering and (3) syntactic rules are incorporated into phrase reordering in decoding step. Thus, it is interesting to investigate the effect of each new feature. IN-SR-MBDecoder is employed to conduct this experiment. Table 3 gives the results. We can see that only distinguishing syntactic phrase reordering from non-syntactic one could obtain a significant improvement over the baseline MEBTG. It has verified our conjecture that syntactic reordering and non-syntactic reordering play different roles and should not be considered the same. On this basis, we integrate the soft rules and the result is promising with 0.69 BLEU improvement. It indicates that the syntactic rules can help phrase reordering in decoding to a large extent. Finally, we add an extra rewarding feature to encourage syntactic phrase reordering. The result shows that this feature can also improve the translation quality. However, our contribution is the combination of the three features as a framework to integrate syntactic rules and the results have shown the effectiveness.

- **Syntactic rules better than lexical ones?**

The key idea in our paper is employing syntactic rules to replace lexical ones if match. We may argue that whether the syntactic rules are indeed more reliable than lexical ones. The experimental results have proved that empirically. And according to our analysis, we find that the syntactic rules are obviously better than lexical ones if the parse tree is correct. For example, the probability $\mathbf{P(i)}$ in soft rule $\langle \mathbf{CP}, \mathbf{NP}, \mathbf{P(i)} \rangle$ in Figure 4 is 0.9796 recommending strong reordering (correct case) while the lexical one predicted with boundary words of phrases is 0.6687. And we also find that when the tree is parsed with error, most syntactic rules in low quality still can be made up for during decoding with translation model and language model. For example, the probability $\mathbf{P(i)}$ of soft rule $\langle \mathbf{PP}, \mathbf{VP}, \mathbf{P(i)} \rangle$ in Figure 3 is 0.6826 which is slightly bigger than 0.6094 of lexical one and so the syntactic rule has a slightly bigger trend to wrong reordering; and however this incorrect rule is made up for in our approach and a similar translation to that of MEBTG (using lexical rules) is obtained as Figure 3 shows. Based on the above analysis, we can conclude that the syntactic rules are better than lexical ones on the whole.

Src:	((迁移)$_{\mathbf{NP}}$ (到$_{\mathrm{P}}$ 新 的 办公 大楼)$_{\mathbf{PP}}$ 将$_{\mathrm{ADVP}}$ (是 一个 挑战)$_{\mathrm{VP}}$)$_{\mathrm{VP}}$
Ref:	relocation to a new office building will be a challenge
MEBTG:	relocation to new office building will be a challenge
MEBTG+HR:	migration will be a challenge to the new office building
HR-IN-MDecoder:	relocation to a new office building will be a challenge

Figure 3: An example that the hard rule is wrong because the **NP** and **PP** are parsed with error, and the reordering system **MEBTG+HRP** leads to a wrong translation which is even worse than the baseline **MEBTG**, but our approach gets a correct translation.

Src:	(首家 (获准 (在 中国)$_{\mathrm{PP}}$ 经营 人民币 业务 的$_{\mathrm{DEC}}$)$_{\mathbf{CP}}$ (比利时 银行)$_{\mathbf{NP}}$)$_{\mathrm{NP}}$
Ref:	the first Belgian bank authorized to operate renminbi business in China
MEBTG:	the first authorized to operate rmb business Belgian bank in China
MEBTG+SR:	the first authorized to operate rmb Belgian bank in China
SR-IN-MDecoder:	the first Belgian bank authorized to operate rmb business in China

Figure 4: An example that the soft rules miss the reordering instance that the **CP** should moved after its sibling **NP**, and the reordering system **MEBTG+SRP** causes a wrong translation just as the baseline **MEBTG** does; however our approach obtains the right one.

6 Conclusion

In this paper, we have presented a framework for effectively incorporating syntactic rules into the phrase-based SMT. For a test sentence to be translated, we first acquire the syntactic reordering rules from the Chinese parse trees. Instead of using them to reorder the source sentences, we incorporate these rules to guide phrase reordering in the decoding stage. To enhance the syntactic phrase reordering, we distinguished the syntactic phrase reordering from non-syntactic one and created an extra binary feature to reward the syntactic reordering. The experiments show that our approach of using syntactic rules significantly outperforms the previous approach whether for hard rules or for soft rules. Furthermore, we have found that just distinguishing syntactic reordering from non-syntactic one could improve the translation quality much and meanwhile facilitate the integration of syntactic rules.

References

Badr, I. 2009. Syntactic Phrase Reordering for English-to-Arabic Statistical Machine Translation. In *Proceedings of EACL 2009.*

Cherry, C. 2008. Cohesive Phrase-based Decoding for Statistical Machine Translation. In *Proceedings of ACL-HLT 2008.*

Collins, M., P. Koehn, and I. Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. In *Proceedings of ACL 2005.*

Habash, N. 2007. Syntactic Preprocessing for Statistical Machine Translation. In *Proceedings of Machine Translation Submit 2007.*

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

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

Li, C., D.D. Zhang, M. Li, M. Zhou, M.H. Li and Y. Guan. 2007. A probabilistic Approach to Syntax-based Reordering for Statistical Machine Translation. In *Proceedings of ACL 2007.*

Marton Y. and P. Resnik. 2008. Soft Syntactic Constrains for Hierarchical Phrase-Based Translation. In *Proceedings of ACL-HLT 2008.*

Och, F.J. and H. Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL.*

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

Stolcke, A. 2002. SRILM- An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Understanding, 2002.*

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

Wu, D.K. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics, 23(3):377-403.*

Xiong, D.Y., Q. Liu and S.X. Lin. 2006. Maximum Entropy based Phrase Reordering Model for Statistical Machine Translation. In *Proceedings of ACL-COLING 2006.*

Xiong, D.Y., M. Zhang, A. Aw and H.Z. Li. 2008. Linguistically Annotated BTG for Statistical Machine Translation. In *Proceedings of COLING 2008.*

Zhang, D.D., M. Li, C.H. Li and M. Zhou. 2007. Phrase Reordering Model Integrating Syntactic Knowledge for SMT. In *Proceedings of EMNLP-CoNLL 2007.*

Zhang, L. 2004. Maximum Entropy Modeling Toolkit for Python and C++. *http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.*