A Generalized Reordering Model for Phrase-Based Statistical Machine Translation

Yanqing He
Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China
yqhe@nlpr.ia.ac.cn

Chengqing Zong
Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China
cqzong@nlpr.ia.ac.cn

Abstract

Phrase-based translation models are widely studied in statistical machine translation (SMT). However, the existing phrase-based translation models either can not deal with non-contiguous phrases or reorder phrases only by the rules without an effective reordering model. In this paper, we propose a generalized reordering model (GREM) for phrase-based statistical machine translation, which is not only able to capture the knowledge on the local and global reordering of phrases, but also is able to obtain some capabilities of phrasal generalization by using non-contiguous phrases. The experimental results have indicated that our model outperforms MEBTG (enhanced BTG with a maximum entropy-based reordering model) and HPTM (hierarchical phrase-based translation model) by improvement of 1.54% and 0.66% in BLEU.

1 Introduction

In statistical machine translation (SMT), phrase-based translation models (Marcu and Wong, 2002; Koehn et al., 2003; Och and Ney, 2004) have advanced word-based model (Brown et al., 1993). In phrase-based translation models, a phrase is possible any contiguous substring without any syntactic constraints, which learns some knowledge on local reordering, translations of multiword expressions, and insertions or deletions that are sensitive to local context. However, some major problems, such as lack of non-contiguous phrases, weak reordering and less generalization ability, are not yet effectively addressed in the existing phrase-based models.

In order to overcome the weakness of existing phrase-based models, two problems must be solved. The first problem is the type of phrases, which may not only involve the contiguous strings but also some non-contiguous strings. The second problem is the reordering of phrases. Bracket transduction grammar (BTG) (Wu, 1995) can reorder two contiguous translations of any two contiguous source strings in straight or inverted directions. BTG is used widely in SMT because of its good tradeoff between efficiency and expressiveness (Zens et al., 2004). Xiong et al. (2006) proposed an enhanced BTG with a maximum entropy-based reordering model (MEBTG). However, in BTG or MEBTG the phrases are only contiguous strings. Such a phrase has no generalization capability. Simard et al. (2005) introduced multi-word expressions into SMT that need not be contiguous in either or both the source and the target side. But in that approach the gap in non-contiguous phrases only stands for a single word.

In this paper, we propose a generalized reordering model for phrase-based statistical machine translation (GREM), which not only properly deals with the non-contiguous phrases but also involves a reordering sub-model of MEBTG. We define a non-contiguous phrase as: $x_1 \Diamond x_2$, which only allows a placeholder $\Diamond$ to connect two contiguous strings x_1 and x_2. Here $\Diamond$ means a gap which can be filled by any contiguous strings. The reason that we only consider such a non-contiguous phrase with only one gap is that this type of phrases has the simplest form considering the efficiency and expressiveness of the model. Under the definition there are four types of phrase pairs: (1) $x \leftrightarrow x$; (2) $x \leftrightarrow x_1 \Diamond x_2$; (3) $x_1 \Diamond x_2 \leftrightarrow x$; (4) $x_1 \Diamond x_2 \leftrightarrow x_1 \Diamond x_2$. Here each type

of phrase pairs permits non-contiguous phrases in either source or target side. In the source side case (1) and case (2) have a contiguous form while case (3) and case (4) have a non-contiguous form. In the target side case (1) and case (3) have a contiguous form while case (2) and case (4) have a non-contiguous form. For a given source contiguous string in the source sentence, we employ some rules to obtain as many target contiguous translations as possible. Our rules (see Section 3) combine non-contiguous or contiguous phrases in source or target side in order to enlarge the candidate translations for the given source string. Then by MEBTG we reorder the target contiguous translations of any two adjacent contiguous phrases in the source sentence to get the final contiguous target translation of the source sentence.

As stated above, our motivations may be explained by the example shown in Figure 1. Given a source sentence in Chinese "在 船 上 我们 有 一位 精通 日语 的 医生", it is supposed that we have eight contiguous or non-contiguous phrase pairs shown in the second row in Figure 1. The arrow denotes the correspondence between the source word and the phrase pair. Each rectangle with round corner means a phrase pair. They have a non-contiguous or contiguous string in the source or target side. In our model, by employing the different combining rules we can obtain some new phrase pairs in the third row. Based on the contiguous phrase pairs in the second row and the third row, reordering rules, such as straight rule or inverted rule, are respectively used to find the correct order of any two adjacent English phrases, and all the English phrases are reordered to get the final target translation: "we have a doctor who can understand Japanese very well in the ship". From the example we can clearly see that our model not only captures the

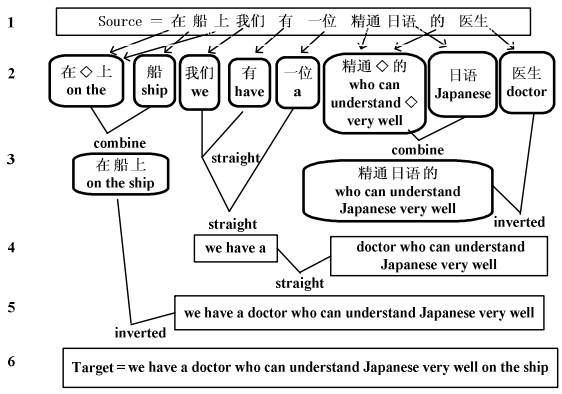

Figure 1: An Example of GREM.

local (by phrase pairs) and global (by MEBTG) reordering of phrases, but also obtains some phrasal generalization by using non-contiguous phrases.

The remainder of the paper is organized as follows: Section 2 gives the related work and Section 3 presents our model of GREM. Afterward we give the implementing process of GREM in Section 4. Section 5 shows the experimental results. Finally, the concluding remarks are given in Section 6.

2 Related work

In order to overcome the weakness of existing phrase-based models, syntax-based models have been proposed and become a hot topic in SMT research. Generally, the syntax-based approaches can be classified into two categories according to the syntactic knowledge source: linguistically syntax-based approaches and formally syntax-based approaches.

The linguistically syntax-based approaches employ syntactic structures informed by syntactic theory. Their syntactic trees are either from the phrase-structured parsers or the dependency parsers (Yamada and Knight, 2001; Galley et al., 2004, 2006; Marcu et al., 2006; Liu et al., 2006; Shieber et al., 1990; Eisner, 2003; Quirk et al., 2005; Ding and Palmer, 2005). All these linguistically syntactic approaches use syntactic structured information to enhance their reordering capability and use some non-contiguous phrases to obtain some generalization. However, these models are highly dependent on the syntactic parsers, and their performances are restricted by the accuracy of syntactic parsers.

The formally syntax-based models are a simple and powerful mechanism to improve the phrase-based approaches, which use synchronous context-free grammar (SCFG) but induce a grammar from a parallel text without relying on any linguistic annotations or assumptions. Chiang (2007) proposed a hierarchical phrase-based translation model (HPTM) that reorganizes phrases into hierarchical ones by reducing sub-phrases to variables. HPTM not only captures the reordering of phrases but also integrates some phrasal generalizations into the global model.

Inspired by the successes of HPTM and MEBTG we propose our GREM model, in which non-contiguous phrases are introduced into MEBTG, thus more generalization capability than MEBTG has been realized. So we have a reorder sub-model based on maximum entropy

for any two contiguous phrases , which is not limited in reordering phrases seen in training data since it is based on features of phrases, not phrase itself (Xiong et al, 2006). HPTM does not have a feature such as the one in our model and so reorders phrases by lying on the rules. Our non-contiguous phrase pairs are similar to some rules in HPTM which have only one variable. However, the rules of HPTM are more numerous than in GREM because our non-contiguous phrases only allow one gap in source or target side while HPTM can have two or more variables in the hierarchical phrase. Simard et al (2005) only allowed one gap to stand for one word, which limited the generalization of non-contiguous phrases. Our model allows the gap to be filled by any contiguous word sequence and we have a larger global and local reordering ability by using the reordering sub-model of MEBTG.

3 The model

For the notational convenience, we use a Generalized Chomsky Normal Form (GCNF) (Melamed et al., 2004) to express our rules. For the terminal rules, we only translate a contiguous source phrase X or non-contiguous source phrase $X(2)$ into their contiguous or non-contiguous translation x or $x_1 \Diamond x_2$. Here, non-terminals appear at the left-hand side (LHS). The contiguous non-terminal X denotes a contiguous source or target phrase, while non-continuous non-terminals are annotated with the number of their contiguous segments, as $X(2)$ in r_2 corresponding to "$x_1 \Diamond x_2$ ". The terminals on the right-hand side (RHS) from r_1 to r_4 are written in columns which express the four types of our phrase alignment.

$$r_1: \quad \begin{matrix} X \\ X \end{matrix} \Rightarrow \begin{pmatrix} x \\ x \end{pmatrix}$$

$$r_2: \quad \begin{matrix} X \\ X(2) \end{matrix} \Rightarrow \begin{pmatrix} x \\ x_1 \Diamond x_2 \end{pmatrix}$$

$$r_3: \quad \begin{matrix} X(2) \\ X \end{matrix} \Rightarrow \begin{pmatrix} x_1 \Diamond x_2 \\ x \end{pmatrix}$$

$$r_4: \quad \begin{matrix} X(2) \\ X(2) \end{matrix} \Rightarrow \begin{pmatrix} x_1 \Diamond x_2 \\ x_1 \Diamond x_2 \end{pmatrix}$$

Non-terminal productions are expressed as the following forms:

$$r_5: \quad \begin{matrix} X \\ X \end{matrix} \Rightarrow \triangleright\triangleleft \begin{matrix} [1,2] \\ [1,2] \end{pmatrix} \begin{pmatrix} X & X \\ X & X \end{pmatrix}$$

$$r_6: \quad \begin{matrix} X \\ X \end{matrix} \Rightarrow \triangleright\triangleleft \begin{matrix} [1,2] \\ [2,1] \end{pmatrix} \begin{pmatrix} X & X \\ X & X \end{pmatrix}$$

$$r_7: \quad \begin{matrix} X \\ X \end{matrix} \Rightarrow \triangleright\triangleleft \begin{matrix} [1,2] \\ [2,1,2] \end{pmatrix} \begin{pmatrix} X & X \\ X & X(2) \end{pmatrix}$$

$$r_8: \quad \begin{matrix} X \\ X \end{matrix} \Rightarrow \triangleright\triangleleft \begin{matrix} [1,2] \\ [1,2,1] \end{pmatrix} \begin{pmatrix} X & X \\ X(2) & X \end{pmatrix}$$

$$r_9: \quad \begin{matrix} X \\ X \end{matrix} \Rightarrow \triangleright\triangleleft \begin{matrix} [2,1,2] \\ [1,2] \end{pmatrix} \begin{pmatrix} X & X(2) \\ X & X \end{pmatrix}$$

$$r_{10}: \quad \begin{matrix} X \\ X \end{matrix} \Rightarrow \triangleright\triangleleft \begin{matrix} [1,2,1] \\ [1,2] \end{pmatrix} \begin{pmatrix} X(2) & X \\ X & X \end{pmatrix}$$

$$r_{11}: \quad \begin{matrix} X \\ X \end{matrix} \Rightarrow \triangleright\triangleleft \begin{matrix} [1,2,1] \\ [1,2,1] \end{pmatrix} \begin{pmatrix} X(2) & X \\ X(2) & X \end{pmatrix}$$

where the non-terminals appear at the left-hand side (LHS) and in parentheses of the right-hand side (RHS). In each row a role template (Melamed et al., 2004) describes the relative order and contiguity of the RHS non-terminals. For example, in the top row of r_5 , $[1,2]$ indicates the order of the two non-terminals is straight. In the bottom row of r_2 , $[2,1]$ indicates the order of two non-terminals is inverted. r_5 and r_6 respectively correspond to straight rule and inverted rule in MEBTG. In the bottom row of r_7 , $[2,1,2]$ indicates that the second non-terminal both precedes and follows the first one. The $\triangleright\triangleleft$ ("join") operator rearranges the non-terminals in each language according to their role template.

During decoding, terminal rules from r_1 to r_4 translate the source phrases into target phrases to generate the four types of phrase pairs. Non-terminal rules from r_7 to r_{11} combine two adjacent phrase pairs contiguous or non-contiguous into a larger contiguous phrase pair by filling a contiguous phrase into the gap of non-contiguous phrase. Then all the phrase pairs are thus changed into contiguous forms. Afterward, just as MEBTG does, non-terminal rules r_5 and r_6 respectively continuously merge two contiguous phrase pairs into a single larger contiguous phrase pair in the straight order or inverted order. The decoding is finished when the whole source sentence is covered.

We use a log-linear model to model the probability of each rule:

$$\Pr(r_k) = \prod_i \phi_i(r_k)^{\lambda_i} \qquad 1 \le k \le 11 \qquad (1)$$

where the ϕ_i is the *i*-th feature defined on rule r_k and λ_i is its weights.

For the non-terminal rules r_5 and r_6, we use the following features:

$$\Pr(r_k) = \Omega^{\lambda_\Omega} \cdot \Delta_{LM}^{\lambda_{LM}} \quad (2)$$

Where, Ω is a probability of applying the rule computed by a maximum entropy classifier, and λ_Ω is the weight; Δ_{LM} is the increment of the language model score and λ_{LM} is its weight. We compute Δ_{LM} by Equations (3) and (4) (Xiong et al., 2006):

$$\Delta_{LM}^{r_5} = LM(x_1^r x_2^l) - LM(x_1^r) - LM(x_2^l) \quad (3)$$

$$\Delta_{LM}^{r_6} = LM(x_2^r x_1^l) - LM(x_2^r) - LM(x_1^l) \quad (4)$$

When we use *n*-gram language model, x_1^l and x_1^r respectively denote the leftmost and the rightmost *n*-1 words of contiguous string x_1, and the corresponding notation of other contiguous string is the same. $LM(\bullet)$ is the log probability of language model of a string $\bullet$. For other rules, we use the following features:

- Translation probabilities in both directions;
- Lexical translation probabilities in both directions;
- Rules penalty;
- Word penalty;
- Language model;

We define the derivation *D* as a sequence of applications of rules from r_1 to r_{11}. Let *c*(*D*) and *e*(*D*) respectively be the Chinese and English yields of *D*. We use a log-linear model to model the probability of a derivation *D*:

$$\Pr(D) = \prod_j \Pr(j) \quad (5)$$

where Pr(*j*) is the probability of the *j*-th application of these rules. Given a source sentence *c*, it finds the final translation *e** generated from the best derivation *D**:

$$e^* = e(D^*) = e(\arg\max_{c(D)=c} \Pr(D)) \quad (6)$$

4 Phrase Extraction and Parameter Training

We start this section with a word-aligned corpus: a set of triple $<c, e, A>$, where *c* is a Chinese sentence, *e* is an English sentence, and *A* is the word alignment between *c* and *e*.

4.1 Phrase extraction

The phrase-based models often obtain phrase pairs satisfying Definition 1 (Och and Ney, 2004; Koehn et al., 2003):

Definition 1. Given a word-aligned sentence pair $<c, e, A>$, $\overline{c}$ or $\overline{e}$ is any contiguous string in sentence *c* or *e*. $<\overline{c}, \overline{e}>$ is a phrase pair iff:

(1) $\forall c_i \in \overline{c} : (i, j) \in A \rightarrow e_j \in \overline{e}$;

(2) $\forall e_j \in \overline{e} : (i, j) \in A \rightarrow c_i \in \overline{c}$.

Based on Definition 1, we extract the phrase pairs satisfying the following Definition 2:

Definition 2. Given a word-aligned sentence pair $<c, e, A>$, $\overline{c}$ or $\overline{e}$ is any contiguous string in sentence *c* or *e*. $\overline{c}_1 \Diamond \overline{c}_2$ is a non-contiguous Chinese string, $\overline{e}_1 \Diamond \overline{e}_2$ is a non-contiguous English string. A Chinese phrase $\tilde{c}$ is either a contiguous Chinese string $\overline{c}$ or a non-contiguous Chinese string $\overline{c}_1 \Diamond \overline{c}_2$, namely $\tilde{c} \in \{\overline{c}_1, \overline{c}_1 \Diamond \overline{c}_2\}$. An English phrase $\tilde{e}$ is either a contiguous English string $\overline{e}$ or a non-contiguous English string $\overline{e}_1 \Diamond \overline{e}_2$, namely $\tilde{e} \in \{\overline{e}, \overline{e}_1 \Diamond \overline{e}_2\}$. Then $<\tilde{c}, \tilde{e}>$ is a phrase pair iff:

(1) $\forall c_i \in \tilde{c} : (i, j) \in A \rightarrow e_j \in \tilde{e}$;

(2) $\forall e_j \in \tilde{e} : (i, j) \in A \rightarrow c_i \in \tilde{c}$.

The above scheme can generate a set of phrase pairs corresponding to (1) $x \leftrightarrow x$; (2) $x \leftrightarrow x_1 \Diamond x_2$; (3) $x_1 \Diamond x_2 \leftrightarrow x$; (4) $x_1 \Diamond x_2 \leftrightarrow x_1 \Diamond x_2$.

The algorithm shown in Figure 2 is designed to extract the first, second and the fourth types of phrase pairs according to the word alignment from *c* to *e*. The variable *PPSet* means the set of phrase pairs extracted. *PPSet_1*, *PPSet_2*, *PPSet_3*, *PPSet_4* respectively denote the set of the above four types of phrase pairs. Inspired by the ideas of hierarchical phrases given by Chiang (2005) we mark the orientation of the gap during extracting process. For the second type phrase pair we mark "CL" or "CR" respectively if the gap in the English side is aligned to the left or right of the Chinese side. For the third type phrase pair we mark "EL" or "ER" if the gap in the Chinese side is aligned to the left or right of the English side. For the first and the fourth type of phrase pairs we don't need such marks. With such a mark we can use our phrase pairs just like some of hierarchical phrases. In the same way we can also get the first, third and fourth type of phrase pairs according to the word alignment from *e* to *c*. Then we merge the phrase pairs from the two directions. For the fourth type we only

Input: sentence pair c, e and word alignment A
Output: *PPSet_1, PPSet_2 and PPSet_4*
1: $PPSet\_1 = \phi$, $PPSet\_2 = \phi$ *and* $PPSet\_4 = \phi$
2: for each span $(i_1, i_2) \in c$ do
3: find phrase pair b= $< c_{i_1}^{i_2}, e_{j_1}^{j_2} >$ that is consistent with A, b $\cup$ *PPSet_1*
4: extend the non-aligned target words around $e_{j_1}^{j_2}$ to get b'= $< c_{i_1}^{i_2}, e_{j'_1}^{j'_2} >$ and b' $\cup$ *PPSet_1*
5: find phrase pair b= $< c_{i_1}^{i_2}, e_{j_1}^{j_3} \Diamond e_{j_4}^{j_2} >$ that is consistent with A, $j_1 \le j_3 < j_4 \le j_2$ and b $\cup$ *PPSet_2*
6: Extend the non-aligned target words around $e_{j_1}^{j_2}$ to get b'= $< c_{i_1}^{i_2}, e_{j'_1}^{j_3} \Diamond e_{j_4}^{j'_2} >$ and b' $\cup$ *PPSet_2*
7: for each $b_1 = < c_{i_1}^{i_2}, e_{j_1}^{j_2} > \in PPSet\_1$
8: for each $b_2 = < c_{i_3}^{i_4}, e_{j_3}^{j_4} > \in PPSet\_1$
9: if(b_1!=b_2 and $j_1 \le j_3 - 1 < j_4 + 1 \le j_2$ and $i_1 \le i_3 - 1 < i_4 + 1 \le i_2$)
10: b= $< c_{i_1}^{i_3-1} \Diamond c_{i_4+1}^{i_2}, e_{j_1}^{j_3-1} \Diamond e_{j_4+1}^{j_2} >$ and b $\cup$ *PPSet_4*

Figure 2: Extraction Algorithm of Contiguous and Non-contiguous Phrase Pairs.

choose the intersection of the two directions in order to increase the precise.

After extraction of phrase pairs, the features of phrase translation are computed as phrase-based translation models (Koehn, 2004). In our training approach we only look the gap $\Diamond$ of non-contiguous phrases as a common word and each phrase pair has four probabilities: two translation probabilities in both directions and two lexical translation probabilities in both directions. We give a count of one to each extracted phrase pair occurrence, and then distribute its weight equally among the contiguous and non-contiguous phrases. Then treating distribution as our observed data to estimate the relative-frequency, we get the phrase translation probabilities in both directions. The lexical translation probabilities are calculated for each bilingual phrase in both directions in the same way as in (Koehn, 2004).

4.2 Maximum Entropy-Based Reordering Model

For reordering of the contiguous phrase pairs we choose the maximum entropy-based reordering model given by Xiong et al. (2006). The model only extracts features from two consecutive phrase pairs and reorders any phrase pairs, regardless of its appearing in the training data or not. We extract reordering example from the word-aligned training corpus and extract the following features from every two consecutive phrase pairs:

- Lexical features: the first word or the last word of two source phrases or target phrases;
- Collocation features: the combination of lexical features.

Then we use these features to train maximum entropy reordering model.

4.3 Decoder

We have developed a bottom-up CKY style decoder. Given a source sentence *c*, we first initiate our search space with our phrase translation table by applying the terminal rules from r_1 to r_4. Each contiguous or non-contiguous source phrase has two possible translation choices: contiguous translations or non-contiguous translations or both of them. All possible derivations spanning from *i* to *j* on the source side are put in the cell of our chart from *i* to *j*. Any sub-cells within (*i*, *j*) have been expanded before cell (*i*, *j*) is expanded. We take two steps to complete the derivation for every sub-cell. The combining step employs rules from r_7 to r_{11} to obtain the initial hypotheses for each sub-cell and the score of the new translation is computed by merging the score of the two sub-derivations. Afterward, in each cell only contiguous translations are saved. The reordering step applies rules r_5 and r_6 which is similar to the translation of BTG. When the whole source sentence is covered, the decoding is finished.

We use three types of pruning strategies: recombination, threshold pruning, and histogram pruning to tradeoff the speed and performance of our decoder. Threshold pruning means that a partial hypothesis that is worse than β times the best score in the same cell is discarded.

5 Experiments

This section gives the results of experiments on a parallel corpus of Chinese-English texts. We carried out the experiments to compare our GREM model against HPTM and MEBTG models.

5.1 Corpus

We use the data of IWSLT07 (International Workshop on Spoken Language Translation 2007) as our experimental data. Table 1 gives the detailed statistics of the experimental data. Our training set contains 39,953 sentence pairs of Chinese-to-English training data released by IWSLT07 and 235,929 sentence pairs from the website[1]. We choose IWSLT07_CE_devset4 released by IWSLT 2007 as our development set to adjust our parameters and test set released by IWSLT 2007 as our test set. In Table 1 'Sent' means *sentence*, 'Voc' denotes the *vocabulary*, and 'A.S.L' is the abbreviation of *average sentence length*.

Set	Language	Sent.	Voc.	A.S.L
Train	Chinese	275,882	11,661	6.2
	English	275,882	12,454	6.7
Dev	Chinese	489	1144	12.8
	English	3,423	2150	13.4
Test	Chinese	489	862	6.5
	English	2,934	1,527	7.7

Table 1: Statistics of Training Set, Development Set and Test Set.

5.2 Experimental results

Bruin and Hiero are the baseline systems for comparison which are respectively implemented in our lab according to Xiong (2006) and Chiang (2005). We obtain the word alignment by running GIZA++ (Och and Ney, 2000) on the training corpus in both directions and applying the "grow-diag-final" refinement rule to get a single many-to-many word alignment for each sentence pair. We built 3-gram language models using the English side of our training set by the SRILM toolkit (Stolcke, 2002).

For the Hiero system initial rules satisfying Definition 1 are extracted and then rule subtraction is performed to obtain rules with no more than two non-terminals. We set a limitation that initial rules are of no more than 9 words and other rules should have no more than 5 terminals and non-terminals. The decoder is CKY-style chart parser that maximizes the derivation probability. The search space is pruned with a chart cell size limitation of 40. Threshold pruning is also used to prune the translation hypotheses that are worse than the current best hypothesis in the cell by a factor of 10.

For Bruin system, we extract the phrase pairs that satisfy Definition 1 given in Section 4.1. The maximum length of source phrase is limited to 9 words. We also extract reordering examples from the training corpus and train the reorder model by the maximum entropy based classifier from the website[2]. During decoding we limit the phrase table within 40 and the partial hypothesis within 200.

In our GREM-based system the main parameters, such as the reorder model, language model, and contiguous phrase table are the same with Bruin except we add a non-contiguous phrase table into our system.

The weights of the different features are trained using the maximum BLEU training algorithm defined by (Venugopal et al. 2005).

In order to compare the numbers of the rules in Hiero system and in our systems, we choose a parallel corpus with human-annotated word alignment which consists of 502 sentence pairs. We extract Hiero rules from the corpus and limit no more than two non-terminals. We set initial rules no more than 10 words and other rules no more than 5 terminals and non-terminals. We get 406,458 rules of Hiero which include one or two variables. Our phrase pair can cover all the Hiero rules with one non-terminal and some rules with two non-terminals. For example, if a Hiero rules with the form like "$<X_1 string X_2 string, X_1 string X_2 string>$", where "string" denotes a terminal string and "X"

Non-terminal	Rule	Percentage (%)
One non-terminal	r_5	8.32
	r_6	1.83
	r_7	7.20
	r_8	6.52
	r_9	3.88
	r_{10}	6.46
	r_{11}	28.36
Two non-terminals	r_5~r_{11}	13.59
Total	76.16	

Table 2: The comparison of Rules Between Hiero and GREM.

[1] http://iwslt07.itc.it/ menu/resources.html

[2] http://maxent.sourceforge.net/

denotes a non-terminal, we can look it as the combination of our r_5 and r_{11}. Table 2 gives the statistics of comparison between Hiero rules and ours. We can see that our rules can cover about 76.16% of Hiero rules.

We also give the numbers of the rules applied in each translation process of the three systems in Table 3. We extract each type of rules from the training data and filter them according to the development set or test set. For Bruin system only contiguous phrases are used. For the other system contiguous and non-contiguous phrases are used. From Table 3 we can see that our rules are much fewer than Hiero system.

System	Filtered By DevSet	Filtered By TestSet
Bruin	157,784	141,347
Hiero	4,192,871	2,612,076
GREM	335,429	184,272

Table 3: The Numbers of the Filtered Rules in Different Systems.

The translation performance of the three systems is shown in Table 4. We can find that our system outperforms the baseline system Bruin and Hiero with a relative improvement of 1.54% and 0.66% in BLEU.

System	BLEU-4	NIST
Bruin	0.3766	6.3844
Hiero	0.3799	6.4293
GREM	0.3824	6.4892

Table 4: The Comparison of Different Systems.

Analysis of our experimental results reveals that our model obtains some generalization of phrases over Bruin by introducing the non-contiguous phrases. Compared with Hiero, our model achieves a comparable performance even with fewer rules. The rules of our model can be considered as a subset of hierarchical phrases of Hiero because our rules only allow one gap while the rules of Hiero have one or more non-terminals. Our experiments also indicate that the large number of rules in Hiero can be simplified to a relative concise form like our rules.

6 Conclusions

In this paper, we propose a generalized reordering model for phrase-based statistical machine translation (GREM). Our model not only captures the local and global reordering of phrases but also obtain some phrasal generalization by using non-contiguous phrases. Our experiments have shown that our novel model outperforms the baseline MEBTG and HPTM by improvement of 1.54% and 0.66% in BLEU.

In the next step, we will incorporate more syntactical features for each rule to get better translation performance.

7 Acknowledgments

The research work described in this paper has been funded by the Natural Science Foundation of China under Grant No. 60575043 and 60736014, the National Key Technology R&D Program under Grant No. 2006BAH03B02, the Hi-Tech R&D Program ("863" Program) of China under Grant No. 2006AA01Z194 and 2006AA010108-4, and Nokia (China) Co. Ltd as well.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. *The mathematics of statistical machine translation: Parameter estimation, Computational Linguistics*, 19(2):263–311.

David Chiang. 2007. *Hierarchical phrase-based translation. Computational Linguistics*, 33(2): 201-228.

Yuan Ding and Martha Palmer. 2005. *Machine translation using probabilistic synchronous dependency insertion grammars*. In proceeding of 43th Meeting of the Association for Computational Linguistics, 541-548

Jason Eisner. 2003. *Learning non-isomorphic tree mappings for machine translation*. In proceedings of the 41th Meeting of the Association for Computational Linguistics (companion volume).

Comeron S. Fordyce. *Overview of the IWSLT 2007 Evaluation Campaign*. In Proceeding of International Workshop on Spoken Language Translation, Trento, Italy, October, 2007.

Michel Galley, Mark Hopkins, Kevin Knight and Daniel Marcu. 2004. *What's in a translation rule?* In proceedings of HLTNAACL- 2004.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, Ignacio Thayer. 2006. *Scalable Inference and Training of Context-Rich Syntactic Translation Models.* In Proceedings of the joint conference of the International Committee on Computational Linguistics

and the Association for Computational Linguistics. Sydney, Australia.

Philipp Koehn, Franz J. Och and Daniel Marcu. 2003. *Statistical phrase-based translation.* In proceedings of HLT-NAACL-03, 127-133

Philipp Koehn. 2004. *Pharaoh: a beam search decoder for phrase-based statistical machine translation models*. In Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas, 115–124.

Yang Liu, Qun Liu and Shouxun Lin. 2006. *Tree-to-String Alignment Template for Statistical Machine Translation.* In proceedings of ACL-06, 609-616.

Daniel Marcu and William Wong. 2002. *A phrase-based, joint probability model for statistical machine translation.* In proceedings of EMNLP-02, 133-139.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. *SPMT: Statistical Machine Translation with Syntactified Target Language Phraases.* In Proceedings of EMNLP-2006, 44-52, Sydney, Australia.

Dan Melamed. 2004. *Statistical machine translation by parsing.* In proceedings of the 42th Meeting of the Association for Computational Linguistics, 653-660

Franz J. Och. 2000. *GIZA++: Training of statistical translation models.* Technical report, RWTH Aachen, University of Technology.

Franz J. Och and Hermann Ney. 2004. *The alignment template approach to statistical machine translation. Computational Linguistics*, 30(4):417-449

Chris Quirk, Arul Menezes and Colin Cherry. 2005. Dependency treelet translation: *Syntactically informed phrasal SMT.* In proceedings of the 43th Meeting of the Association for Computational Linguistics, 271-279

S. Shieber and Y. Schabes. 1990. *Synchronous tree adjoining grammars.* In proceedings of COLING-90.

Andreas Stolcke, 2002. *SRILM – an extensible language modeling toolkit.* In proceedings of International Conferrence on spoken Language processing, 2:901-904

Simard, Michel, Nicola Cancedda, Bruno Cavestro, Marc DymetMan, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlasi , and Arne mauser, 2005. *Translation with Non-contiguous phrases.* In Proceedings of HLT/EMNLP 2005, pages 755-762, Ann Arbor, MI.

Ashish Venugopal, Stephan Vogel and Alex Waibel. 2003. *Effective Phrase Translation Extraction from Alignment Models*, in Proceedings of the 41st ACL, 319-326.

Dekai Wu. 1995. *Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora*. In proceeding of IJCAL 1995, 1328-1334,Montreal, August.

Dekai Wu. 1997. *Stochastic inversion transduction grammars and bilingual parsing of parallel corpora.* Computational Linguistics, 23(3):377-403

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. *Maximum Entropy Based phrase reordering model for statistical machine translation.* In proceedings of COLING-ACL, Sydney, Australia.

Kenji Yamada and Kevin Knight. 2001. *A syntax-based statistical translation model*. In proceedings of the 39th Meeting of the ACL, 523-530.

R. Zens, H. Ney, T. Watanabe, and E. Sumita. 2004. *Reordering Constraints for Phrase-Based Statistical Machine Translation.* In Proceedings of CoLing 2004, Geneva, Switzerland, pp. 205-211.