Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation

Deyi Xiong
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China, 100080
Graduate School of Chinese Academy of Sciences
dyxiong@ict.ac.cn

Qun Liu and Shouxun Lin
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China, 100080
{liuqun, sxlin}@ict.ac.cn

Abstract

We propose a novel reordering model for phrase-based statistical machine translation (SMT) that uses a maximum entropy (MaxEnt) model to predicate reorderings of neighbor blocks (phrase pairs). The model provides content-dependent, hierarchical phrasal reordering with generalization based on features automatically learned from a real-world bitext. We present an algorithm to extract all reordering events of neighbor blocks from bilingual data. In our experiments on Chinese-to-English translation, this MaxEnt-based reordering model obtains significant improvements in BLEU score on the NIST MT-05 and IWSLT-04 tasks.

1 Introduction

Phrase reordering is of great importance for phrase-based SMT systems and becoming an active area of research recently. Compared with word-based SMT systems, phrase-based systems can easily address reorderings of words within phrases. However, at the phrase level, reordering is still a computationally expensive problem just like reordering at the word level (Knight, 1999).

Many systems use very simple models to reorder phrases [1]. One is distortion model (Och and Ney, 2004; Koehn et al., 2003) which penalizes translations according to their jump distance instead of their content. For example, if N words are skipped, a penalty of N will be paid regardless of which words are reordered. This model takes the risk of penalizing long distance jumps which are common between two languages with very different orders. Another simple model is flat reordering model (Wu, 1996; Zens et al., 2004; Kumar et al., 2005) which is not content dependent either. Flat model assigns constant probabilities for monotone order and non-monotone order. The two probabilities can be set to prefer monotone or non-monotone orientations depending on the language pairs.

In view of content-independency of the distortion and flat reordering models, several researchers (Och et al., 2004; Tillmann, 2004; Kumar et al., 2005; Koehn et al., 2005) proposed a more powerful model called lexicalized reordering model that is phrase dependent. Lexicalized reordering model learns local orientations (monotone or non-monotone) with probabilities for each bilingual phrase from training data. During decoding, the model attempts to finding a Viterbi local orientation sequence. Performance gains have been reported for systems with lexicalized reordering model. However, since reorderings are related to concrete phrases, researchers have to design their systems carefully in order not to cause other problems, e.g. the data sparseness problem.

Another smart reordering model was proposed by Chiang (2005). In his approach, phrases are reorganized into hierarchical ones by reducing subphrases to variables. This template-based scheme not only captures the reorderings of phrases, but also integrates some phrasal generalizations into the global model.

In this paper, we propose a novel solution for phrasal reordering. Here, under the ITG constraint (Wu, 1997; Zens et al., 2004), we need to consider just two kinds of reorderings, *straight* and *inverted* between two consecutive blocks. Therefore reordering can be modelled as a problem of

[1] In this paper, we focus our discussions on phrases that are not necessarily aligned to syntactic constituent boundary.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 521–528,
Sydney, July 2006.

classification with only two labels, *straight* and *inverted*. In this paper, we build a maximum entropy based classification model as the reordering model. Different from lexicalized reordering, we do not use the whole block as reordering evidence, but only features extracted from blocks. This is more flexible. It makes our model reorder any blocks, observed in training or not. The whole maximum entropy based reordering model is embedded inside a log-linear phrase-based model of translation. Following the Bracketing Transduction Grammar (BTG) (Wu, 1996), we built a CKY-style decoder for our system, which makes it possible to reorder phrases hierarchically.

To create a maximum entropy based reordering model, the first step is learning reordering examples from training data, similar to the lexicalized reordering model. But in our way, any evidences of reorderings will be extracted, not limited to reorderings of bilingual phrases of length less than a predefined number of words. Secondly, features will be extracted from reordering examples according to feature templates. Finally, a maximum entropy classifier will be trained on the features.

In this paper we describe our system and the MaxEnt-based reordering model with the associated algorithm. We also present experiments that indicate that the MaxEnt-based reordering model improves translation significantly compared with other reordering approaches and a state-of-the-art distortion-based system (Koehn, 2004).

2 System Overview

2.1 Model

Under the BTG scheme, translation is more like monolingual parsing through derivations. Throughout the translation procedure, three rules are used to derive the translation

$$A \xrightarrow{[\,]} (A^1, A^2) \tag{1}$$

$$A \xrightarrow{\langle\,\rangle} (A^1, A^2) \tag{2}$$

$$A \rightarrow (x, y) \tag{3}$$

During decoding, the source sentence is segmented into a sequence of phrases as in a standard phrase-based model. Then the lexical rule (3) [2] is used to translate source phrase y into target phrase x and generate a block A. Later, the *straight* rule (1) merges two consecutive blocks into a single larger block in the straight order; while the *inverted* rule (2) merges them in the inverted order. These two merging rules will be used continuously until the whole source sentence is covered. When the translation is finished, a tree indicating the hierarchical segmentation of the source sentence is also produced.

In the following, we will define the model in a straight way, not in the dynamic programming recursion way used by (Wu, 1996; Zens et al., 2004). We focus on defining the probabilities of different rules by separating different features (including the language model) out from the rule probabilities and organizing them in a log-linear form. This straight way makes it clear how rules are used and what they depend on.

For the two merging rules *straight* and *inverted*, applying them on two consecutive blocks A^1 and A^2 is assigned a probability $Pr^m(A)$

$$Pr^m(A) = \Omega^{\lambda_\Omega} \cdot \triangle_{p_{LM}(A^1,A^2)}^{\lambda_{LM}} \tag{4}$$

where the Ω is the reordering score of block A^1 and A^2, λ_Ω is its weight, and $\triangle_{p_{LM}(A^1,A^2)}$ is the increment of the language model score of the two blocks according to their final order, λ_{LM} is its weight.

For the lexical rule, applying it is assigned a probability $Pr^l(A)$

$$\begin{aligned} Pr^l(A) \;=\; & p(x|y)^{\lambda_1} \cdot p(y|x)^{\lambda_2} \cdot p_{lex}(x|y)^{\lambda_3} \\ & \cdot p_{lex}(y|x)^{\lambda_4} \cdot exp(1)^{\lambda_5} \cdot exp(|x|)^{\lambda_6} \\ & \cdot p_{LM}^{\lambda_{LM}}(x) \end{aligned} \tag{5}$$

where $p(\cdot)$ are the phrase translation probabilities in both directions, $p_{lex}(\cdot)$ are the lexical translation probabilities in both directions, and $exp(1)$ and $exp(|x|)$ are the phrase penalty and word penalty, respectively. These features are very common in state-of-the-art systems (Koehn et al., 2005; Chiang, 2005) and λs are weights of features.

For the reordering model Ω, we define it on the two consecutive blocks A^1 and A^2 and their order $o \in \{straight, inverted\}$

$$\Omega = f(o, A^1, A^2) \tag{6}$$

Under this framework, different reordering models can be designed. In fact, we defined four reordering models in our experiments. The first one

[2] Currently, we restrict phrases x and y not to be null. Therefore neither deletion nor insertion is carried out during decoding. However, these operations are to be considered in our future version of model.

is *NONE*, meaning no explicit reordering features at all. We set Ω to 1 for all different pairs of blocks and their orders. So the phrasal reordering is totally dependent on the language model. This model is obviously different from the monotone search, which does not use the *inverted* rule at all. The second one is a distortion style reordering model, which is formulated as

$$\Omega = \begin{cases} exp(0), & o = straight \\ exp(|A^1|) + (|A^2|), & o = inverted \end{cases}$$

where $|A^i|$ denotes the number of words on the source side of blocks. When $\lambda_\Omega < 0$, this design will penalize those non-monotone translations. The third one is a flat reordering model, which assigns probabilities for the straight and inverted order. It is formulated as

$$\Omega = \begin{cases} p_m, & o = straight \\ 1 - p_m, & o = inverted \end{cases}$$

In our experiments on Chinese-English tasks, the probability for the straight order is set at $p_m = 0.95$. This is because word order in Chinese and English is usually similar. The last one is the maximum entropy based reordering model proposed by us, which will be described in the next section.

We define a derivation D as a sequence of applications of rules $(1) - (3)$, and let $c(D)$ and $e(D)$ be the Chinese and English yields of D. The probability of a derivation D is

$$Pr(D) = \prod_i Pr(i) \tag{7}$$

where $Pr(i)$ is the probability of the ith application of rules. Given an input sentence c, the final translation e^* is derived from the best derivation D^*

$$\begin{aligned} D^* &= \operatorname*{argmax}_{c(D)=c} Pr(D) \\ e^* &= e(D^*) \end{aligned} \tag{8}$$

2.2 Decoder

We developed a CKY style decoder that employs a beam search algorithm, similar to the one by Chiang (2005). The decoder finds the best derivation that generates the input sentence and its translation. From the best derivation, the best English e^* is produced.

Given a source sentence c, firstly we initiate the chart with phrases from phrase translation table by applying the lexical rule. Then for each cell that spans from i to j on the source side, all possible derivations spanning from i to j are generated. Our algorithm guarantees that any sub-cells within (i, j) have been expanded before cell (i, j) is expanded. Therefore the way to generate derivations in cell (i, j) is to merge derivations from any two neighbor sub-cells. This combination is done by applying the *straight* and *inverted* rules. Each application of these two rules will generate a new derivation covering cell (i, j). The score of the new generated derivation is derived from the scores of its two sub-derivations, reordering model score and the increment of the language model score according to the Equation (4). When the whole input sentence is covered, the decoding is over.

Pruning of the search space is very important for the decoder. We use three pruning ways. The first one is recombination. When two derivations in the same cell have the same w leftmost/rightmost words on the English yields, where w depends on the order of the language model, they will be recombined by discarding the derivation with lower score. The second one is the threshold pruning which discards derivations that have a score worse than α times the best score in the same cell. The last one is the histogram pruning which only keeps the top n best derivations for each cell. In all our experiments, we set $n = 40, \alpha = 0.5$ to get a tradeoff between speed and performance in the development set.

Another feature of our decoder is the k-best list generation. The k-best list is very important for the minimum error rate training (Och, 2003a) which is used for tuning the weights λ for our model. We use a very lazy algorithm for the k-best list generation, which runs two phases similarly to the one by Huang et al. (2005). In the first phase, the decoder runs as usual except that it keeps some information of weaker derivations which are to be discarded during recombination. This will generate not only the first-best of final derivation but also a shared forest. In the second phase, the lazy algorithm runs recursively on the shared forest. It finds the second-best of the final derivation, which makes its children to find their second-best, and children's children's second-best, until the leaf node's second-best. Then it finds the third-best, forth-best, and so on. In all our experiments, we set $k = 200$.

The decoder is implemented in C++. Using the pruning settings described above, without the k-best list generation, it takes about 6 seconds to translate a sentence of average length 28.3 words on a 2GHz Linux system with 4G RAM memory.

3 Maximum Entropy Based Reordering Model

In this section, we discuss how to create a maximum entropy based reordering model. As described above, we defined the reordering model Ω on the three factors: order o, block A^1 and block A^2. The central problem is, given two neighbor blocks A^1 and A^2, how to predicate their order $o \in \{straight, inverted\}$. This is a typical problem of two-class classification. To be consistent with the whole model, the conditional probability $p(o|A^1, A^2)$ is calculated. A simple way to compute this probability is to take counts from the training data and then to use the maximum likelihood estimate (MLE)

$$p(o|A^1, A^2) = \frac{Count(o, A^1, A^2)}{Count(A^1, A^2)} \quad (9)$$

The similar way is used by lexicalized reordering model. However, in our model this way can't work because blocks become larger and larger due to using the merging rules, and finally unseen in the training data. This means we can not use blocks as direct reordering evidences.

A good way to this problem is to use features of blocks as reordering evidences. Good features can not only capture reorderings, avoid sparseness, but also integrate generalizations. It is very straight to use maximum entropy model to integrate features to predicate reorderings of blocks. Under the MaxEnt model, we have

$$\Omega = p_\theta(o|A^1, A^2) = \frac{exp(\sum_i \theta_i h_i(o, A^1, A^2))}{\sum_o exp(\sum_i \theta_i h_i(o, A^1, A^2))} \quad (10)$$

where the functions $h_i \in \{0, 1\}$ are model features and the θ_i are weights of the model features which can be trained by different algorithms (Malouf, 2002).

3.1 Reordering Example Extraction Algorithm

The input for the algorithm is a bilingual corpus with high-precision word alignments. We obtain the word alignments using the way of Koehn et al. (2005). After running GIZA++ (Och and Ney, 2000) in both directions, we apply the "grow-diag-final" refinement rule on the intersection alignments for each sentence pair.

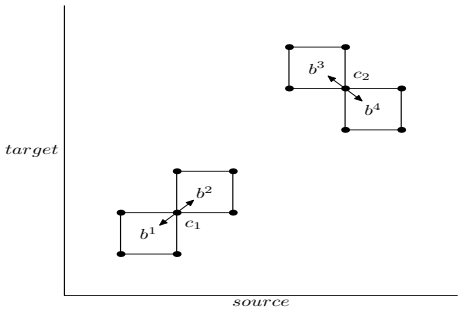

Figure 1: The bold dots are corners. The arrows from the corners are their links. Corner c_1 is shared by block b^1 and b^2, which in turn are linked by the STRAIGHT links, *bottomleft* and *topright* of c_1. Similarly, block b^3 and b^4 are linked by the INVERTED links, *topleft* and *bottomright* of c_2.

Before we introduce this algorithm, we introduce some formal definitions. The first one is *block* which is a pair of source and target contiguous sequences of words

$$b = (s_{i_1}^{i_2}, t_{j_1}^{j_2})$$

b must be consistent with the word alignment M

$$\forall (i, j) \in M, i_1 \leq i \leq i_2 \leftrightarrow j_1 \leq j \leq j_2$$

This definition is similar to that of bilingual phrase except that there is no length limitation over block. A *reordering example* is a triple of (o, b^1, b^2) where b^1 and b^2 are two neighbor blocks and o is the order between them. We define each vertex of block as *corner*. Each corner has four *links* in four directions: *topright*, *topleft*, *bottomright*, *bottomleft*, and each link links a set of blocks which have the corner as their vertex. The *topright* and *bottomleft* link blocks with the straight order, so we call them *STRAIGHT* links. Similarly, we call the *topleft* and *bottomright INVERTED* links since they link blocks with the inverted order. For convenience, we use $b \hookleftarrow \mathcal{L}$ to denote that block b is linked by the link $\mathcal{L}$. Note that the STRAIGHT links can not coexist with the INVERTED links. These definitions are illustrated in Figure 1.

The reordering example extraction algorithm is shown in Figure 2. The basic idea behind this algorithm is to register all neighbor blocks to the associated links of corners which are shared by them. To do this, we keep an array to record link

1: **Input**: sentence pair (s,t) and their alignment M
2: $\Re := \emptyset$
3: **for** each span $(i_1, i_2) \in s$ **do**
4: find block $b = (s_{i_1}^{i_2}, t_{j_1}^{j_2})$ that is consistent with M
5: Extend block b on the target boundary with one possible non-aligned word to get blocks $E(b)$
6: **for** each block $b^* \in b \bigcup E(b)$ **do**
7: Register b^* to the links of four corners of it
8: **end for**
9: **end for**
10: **for** each corner $\mathcal{C}$ in the matrix M **do**
11: **if** STRAIGHT links exist **then**
12: $\Re := \Re \bigcup \{(straight, b^1, b^2)\}$, $b^1 \hookleftarrow \mathcal{C}.bottomleft, b^2 \hookleftarrow \mathcal{C}.topright$
13: **else if** INVERTED links exist **then**
14: $\Re := \Re \bigcup \{(inverted, b^1, b^2)\}$, $b^1 \hookleftarrow \mathcal{C}.topleft, b^2 \hookleftarrow \mathcal{C}.bottomright$
15: **end if**
16: **end for**
17: **Output**: reordering examples $\Re$

Figure 2: Reordering Example Extraction Algorithm.

information of corners when extracting blocks. Line 4 and 5 are similar to the phrase extraction algorithm by Och (2003b). Different from Och, we just extend one word which is aligned to null on the boundary of target side. If we put some length limitation over the extracted blocks and output them, we get bilingual phrases used in standard phrase-based SMT systems and also in our system. Line 7 updates all links associated with the current block. You can attach the current block to each of these links. However this will increase reordering examples greatly, especially those with the $straight$ order. In our Experiments, we just attach the smallest blocks to the STRAIGHT links, and the largest blocks to the INVERTED links. This will keep the number of reordering examples acceptable but without performance degradation. Line 12 and 14 extract reordering examples.

3.2 Features

With the extracted reordering examples, we can obtain features for our MaxEnt-based reordering model. We design two kinds of features, lexical features and collocation features. For a block $b = (s,t)$, we use s_1 to denote the first word of the source s, t_1 to denote the first word of the target t.

Lexical features are defined on the single word s_1 or t_1. Collocation features are defined on the combination s_1 or t_1 between two blocks b^1 and b^2. Three kinds of combinations are used. The first one is source collocation, $b^1.s_1 \& b^2.s_1$. The second is target collocation, $b^1.t_1 \& b^2.t_1$. The last one

$$h_i(o, b^1, b^2) = \begin{cases} 1, & b^1.t_1 = E_1, o = O \\ 0, & otherwise \end{cases}$$

$$h_j(o, b^1, b^2) = \begin{cases} 1, & b^1.t_1 = E_1, b^2.t_1 = E_2, o = O \\ 0, & otherwise \end{cases}$$

Figure 3: MaxEnt-based reordering feature templates. The first one is a lexical feature, and the second one is a target collocation feature, where E_i are English words, $O \in \{straight, inverted\}$.

is block collocation, $b^1.s_1 \& b^1.t_1$ and $b^2.s_1 \& b^2.t_1$. The templates for the lexical feature and the collocation feature are shown in Figure 3.

Why do we use the first words as features? These words are nicely at the boundary of blocks. One of assumptions of phrase-based SMT is that phrase cohere across two languages (Fox, 2002), which means phrases in one language tend to be moved together during translation. This indicates that boundary words of blocks may keep information for their movements/reorderings. To test this hypothesis, we calculate the information gain ratio (IGR) for boundary words as well as the whole blocks against the order on the reordering examples extracted by the algorithm described above. The IGR is the measure used in the decision tree learning to select features (Quinlan, 1993). It represents how precisely the feature predicate the class. For feature f and class c, the $IGR(f,c)$

$$IGR(f,c) = \frac{En(c) - En(c|f)}{En(f)} \qquad (11)$$

where $En(\cdot)$ is the entropy and $En(\cdot|\cdot)$ is the conditional entropy. To our surprise, the IGR for the four boundary words $(IGR(\langle b^1.s_1,\ b^2.s_1,\ b^1.t_1,\ b^2.t_1\rangle,\ order) = 0.2637)$ is very close to that for the two blocks together $(IGR(\langle b^1,\ b^2\rangle,\ order) = 0.2655)$. Although our reordering examples do not cover all reordering events in the training data, this result shows that boundary words do provide some clues for predicating reorderings.

4 Experiments

We carried out experiments to compare against various reordering models and systems to demonstrate the competitiveness of MaxEnt-based reordering:

1. Monotone search: the *inverted* rule is not used.

2. Reordering variants: the *NONE*, distortion and flat reordering models described in Section 2.1.

3. Pharaoh: A state-of-the-art distortion-based decoder (Koehn, 2004).

4.1 Corpus

Our experiments were made on two Chinese-to-English translation tasks: NIST MT-05 (news domain) and IWSLT-04 (travel dialogue domain).

NIST MT-05. In this task, the bilingual training data comes from the FBIS corpus with 7.06M Chinese words and 9.15M English words. The trigram language model training data consists of English texts mostly derived from the English side of the UN corpus (catalog number LDC2004E12), which totally contains 81M English words. For the efficiency of minimum error rate training, we built our development set using sentences of length at most 50 characters from the NIST MT-02 evaluation test data.

IWSLT-04. For this task, our experiments were carried out on the small data track. Both the bilingual training data and the trigram language model training data are restricted to the supplied corpus, which contains 20k sentences, 179k Chinese words and 157k English words. We used the CSTAR 2003 test set consisting of 506 sentence pairs as development set.

4.2 Training

We obtained high-precision word alignments using the way described in Section 3.1. Then we ran our reordering example extraction algorithm to output blocks of length at most 7 words on the Chinese side together with their internal alignments. We also limited the length ratio between the target and source language ($max(|s|,|t|)/min(|s|,|t|)$) to 3. After extracting phrases, we calculated the phrase translation probabilities and lexical translation probabilities in both directions for each bilingual phrase.

For the minimum-error-rate training, we reimplemented Venugopal's trainer [3] (Venugopal et al., 2005) in C++. For all experiments, we ran this trainer with the decoder iteratively to tune the weights λs to maximize the BLEU score on the development set.

[3]See http://www.cs.cmu.edu/ ashishv/mer.html. This is a Matlab implementation.

Pharaoh

We shared the same phrase translation tables between Pharaoh and our system since the two systems use the same features of phrases. In fact, we extracted more phrases than Pharaoh's trainer with its default settings. And we also used our reimplemented trainer to tune lambdas of Pharaoh to maximize its BLEU score. During decoding, we pruned the phrase table with $b = 100$ (default 20), pruned the chart with $n = 100, \alpha = 10^{-5}$ (default setting), and limited distortions to 4 (default 0).

MaxEnt-based Reordering Model

We firstly ran our reordering example extraction algorithm on the bilingual training data without any length limitations to obtain reordering examples and then extracted features from these examples. In the task of NIST MT-05, we obtained about 2.7M reordering examples with the straight order, and 367K with the inverted order, from which 112K lexical features and 1.7M collocation features after deleting those with one occurrence were extracted. In the task of IWSLT-04, we obtained 79.5k reordering examples with the straight order, 9.3k with the inverted order, from which 16.9K lexical features and 89.6K collocation features after deleting those with one occurrence were extracted. Finally, we ran the MaxEnt toolkit by Zhang [4] to tune the feature weights. We set iteration number to 100 and Gaussian prior to 1 for avoiding overfitting.

4.3 Results

We dropped unknown words (Koehn et al., 2005) of translations for both tasks before evaluating their BLEU scores. To be consistent with the official evaluation criterions of both tasks, case-sensitive BLEU-4 scores were computed For the NIST MT-05 task and case-insensitive BLEU-4 scores were computed for the IWSLT-04 task [5]. Experimental results on both tasks are shown in Table 1. Italic numbers refer to results for which the difference to the best result (indicated in bold) is not statistically significant. For all scores, we also show the 95% confidence intervals computed using Zhang's significant tester (Zhang et al., 2004) which was modified to conform to NIST's

[4]See http://homepages.inf.ed.ac.uk/s0450736 /maxent_toolkit.html.

[5]Note that the evaluation criterion of IWSLT-04 is not totally matched since we didn't remove punctuation marks.

definition of the BLEU brevity penalty.

We observe that if phrasal reordering is totally dependent on the language model (*NONE*) we get the worst performance, even worse than the monotone search. This indicates that our language models were not strong to discriminate between straight orders and inverted orders. The flat and distortion reordering models (Row 3 and 4) show similar performance with Pharaoh. Although they are not dependent on phrases, they really reorder phrases with penalties to wrong orders supported by the language model and therefore outperform the monotone search. In row 6, only lexical features are used for the MaxEnt-based reordering model; while row 7 uses lexical features and collocation features. On both tasks, we observe that various reordering approaches show similar and stable performance ranks in different domains and the MaxEnt-based reordering models achieve the best performance among them. Using all features for the MaxEnt model (lex + col) is marginally better than using only lex features (lex).

Systems	NIST MT-05	IWSLT-04
monotone	20.1 ± 0.8	37.8 ± 3.2
NONE	19.6 ± 0.8	36.3 ± 2.9
Distortion	20.9 ± 0.8	38.8 ± 3.0
Flat	20.5 ± 0.8	38.7 ± 2.8
Pharaoh	20.8 ± 0.8	38.9 ± 3.3
MaxEnt (lex)	*22.0* ± 0.8	*42.4* ± 3.3
MaxEnt (lex + col)	**22.2** ± 0.8	**42.8** ± 3.3

Table 1: BLEU-4 scores (%) with the 95% confidence intervals. Italic numbers refer to results for which the difference to the best result (indicated in bold) is not statistically significant.

4.4 Scaling to Large Bitexts

In the experiments described above, collocation features do not make great contributions to the performance improvement but make the total number of features increase greatly. This is a problem for MaxEnt parameter estimation if it is scaled to large bitexts. Therefore, for the integration of MaxEnt-based phrase reordering model in the system trained on large bitexts, we remove collocation features and only use lexical features from the last words of blocks (similar to those from the first words of blocks with similar performance). This time the bilingual training data contain 2.4M sentence pairs (68.1M Chinese words and 73.8M English words) and two trigram language models are used. One is trained on the English side of the bilingual training data. The other is trained on the Xinhua portion of the Gigaword corpus with 181.1M words. We also use some rules to translate numbers, time expressions and Chinese person names. The new Bleu score on NIST MT-05 is 0.291 which is very promising.

5 Discussion and Future Work

In this paper we presented a MaxEnt-based phrase reordering model for SMT. We used lexical features and collocation features from boundary words of blocks to predicate reorderings of neighbor blocks. Experiments on standard Chinese-English translation tasks from two different domains showed that our method achieves a significant improvement over the distortion/flat reordering models.

Traditional distortion/flat-based SMT translation systems are good for learning phrase translation pairs, but learn nothing for phrasal reorderings from real-world data. This is our original motivation for designing a new reordering model, which can learn reorderings from training data just like learning phrasal translations. Lexicalized reordering model learns reorderings from training data, but it binds reorderings to individual concrete phrases, which restricts the model to reorderings of phrases seen in training data. On the contrary, the MaxEnt-based reordering model is not limited by this constraint since it is based on features of phrase, not phrase itself. It can be easily generalized to reorder unseen phrases provided that some features are fired on these phrases.

Another advantage of the MaxEnt-based reordering model is that it can take more features into reordering, even though they are non-independent. Tillmann et. al (2005) also use a MaxEnt model to integrate various features. The difference is that they use the MaxEnt model to predict not only orders but also blocks. To do that, it is necessary for the MaxEnt model to incorporate real-valued features such as the block translation probability and the language model probability. Due to the expensive computation, a local model is built. However, our MaxEnt model is just a module of the whole log-linear model of translation which uses its score as a real-valued feature. The modularity afforded by this design does not incur any computation problems, and make it eas-

ier to update one sub-model with other modules unchanged.

Beyond the MaxEnt-based reordering model, another feature deserving attention in our system is the CKY style decoder which observes the ITG. This is different from the work of Zens et. al. (2004). In their approach, translation is generated linearly, word by word and phrase by phrase in a traditional way with respect to the incorporation of the language model. It can be said that their decoder did not violate the ITG constraints but not that it observed the ITG. The ITG not only decreases reorderings greatly but also makes reordering hierarchical. Hierarchical reordering is more meaningful for languages which are organized hierarchically. From this point, our decoder is similar to the work by Chiang (2005).

The future work is to investigate other valuable features, e.g. binary features that explain blocks from the syntactical view. We think that there is still room for improvement if more contributing features are used.

Acknowledgements

This work was supported in part by National High Technology Research and Development Program under grant #2005AA114140 and National Natural Science Foundation of China under grant #60573188. Special thanks to Yajuan Lü for discussions of the manuscript of this paper and three anonymous reviewers who provided valuable comments.

References

Ashish Venugopal, Stephan Vogel. 2005. Considerations in Maximum Mutual Information and Minimum Classification Error training for Statistical Machine Translation. In the *Proceedings of EAMT-05*, Budapest, Hungary May 30-31.

Christoph Tillmann. 2004. A block orientation model for statistical machine translation. In *HLT-NAACL*, Boston, MA, USA.

Christoph Tillmann and Tong Zhang. 2005. A Localized Prediction Model for statistical machine translation. In *Proceedings of ACL 2005*, pages 557 – 564.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL 2005*, pages 263 – 270.

Dekai Wu. 1996. A Polynomial-Time Algorithm for Statistical Machine Translation. In *Proceedings of ACL 1996*.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 23:377 – 404.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL 2000*, pages 440 – 447.

Franz Josef Och. 2003a. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*, pages 160 – 167.

Franz Josef Och. 2003b. Statistical Machine Translation: From Single-Word Models to Alignment Templates Thesis.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. Computational Linguistics, 30:417 – 449.

Franz Josef Och, Ignacio Thayer, Daniel Marcu, Kevin Knight, Dragos Stefan Munteanu, Quamrul Tipu, Michel Galley, and Mark Hopkins. 2004. Arabic and Chinese MT at USC/ISI. Presentation given at NIST Machine Translation Evaluation Workshop.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of EMNLP 2002*.

J. R. Quinlan. 1993. C4.5: progarms for machine learning. Morgan Kaufmann Publishers.

Kevin Knight. 1999. Decoding complexity in wordreplacement translation models. Computational Linguistics, Squibs & Discussion, 25(4).

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Vancouver, October, pages 53 – 64.

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

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*, pages 115 – 124.

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

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.

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

Shankar Kumar and William Byrne. 2005. Local phrase reordering models for statistical machine translation. In *Proceedings of HLT-EMNLP*.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of LREC 2004*, pages 2051 – 2054.