# Multiple Reorderings in Phrase-based Machine Translation

**Niyu Ge,  Abe Ittycheriah**
IBM T.J.Watson Research
1101 Kitchawan Rd.
Yorktown Heights, NY 10598
(niyuge, abei)@us.ibm.com

**Kishore Papineni**
Yahoo! Research
45 West 18th St.
New York, NY 10011
kpapi@yahoo-inc.com

## Abstract

This paper presents a method to integrate multiple reordering strategies in phrase-based statistical machine translation. Recently there has been much research effort in reordering problems in machine translation. State-of-the-art decoders incorporate sophisticated local reordering strategies, but there is little research on a unified approach to incorporate various kinds of reordering methods. We present a phrase-based decoder which easily allows multiple reordering schemes. We show how to use this framework to perform distance-based reordering and HIERO-style (Chiang 2005) hierarchical reordering. We also present two novel syntax-based reordering methods, one built on part-of-speech tags and the other based on parse trees. We will give experimental results using these relatively easy to implement methods on standard tests.

## 1    Introduction and Previous Work

Given an input source sentence and guided by a translation model, language model, distortion model, etc., a machine translation decoder searches for a target sentence that is the best translation of the source. There are usually two aspects of the search. One tries to find target words for a given source segment. The other searches for the order in which the source segments are to be translated. A source *segment* here means a contiguous part of the source sentence. The former is largely controlled by language models and translation models and the

latter by language models and distortion models. It is, in most cases, the latter, the search for the correct word order (which source segment to be translated next) that results in a large combinatorial search space. State-of-the-art decoders use dynamic programming based beam-search with local reordering (Och 1999, Tillmann 2000). Although local reordering to some degree is implicit in phrase-based decoding, the kind of reordering is very limited. The simplest distance-based reordering, from the current source position $i$, tries to defer the translation of the next $n$ words ($1 \le n \le N$, N the maximum number of words to be delayed). N is bounded by the computational requirements.

Recent work on reordering has been on trying to find "smart" ways to decide word order, using syntactic features such as POS tags (Lee and Ge 2005) , parse trees (Zhang et.al, 2007, Wang et.al. 2007, Collins et.al. 2005, Yamada and Knight 2001) to name just a few, and synchronized CFG (Wu 1997, Chiang 2005), again to name just a few. These efforts have shown promising improvements in translation quality. However, to use these features during decoding requires either a separate decoder to be written or some ad-hoc mechanisms to be invented to incorporate them into an existing decoder, or in some cases (Wang et. al. 2007) the input source is pre-ordered to be decoded monotonically.

(Kanthak et. al. 2005) described a framework in which different reordering methods are represented as search constraints to a finite state automata. It is able to compute distance-based and ITG-style reordering automata. We differ from that approach in a couple of ways. One is that in (Kanthak et. al. 2005), an on-demand

reordering graph is pre-computed which is then taken as a input for monotonic decoding. We compute the reordering as the sentence is being decoded. The second is that it is not clear how to generate the permutation graphs under, say HIERO-type hierarchical constraints, or other syntax-inspired reorderings such as those based on part-of-speech patterns. Our approach differs in that we allow greater flexibility in capturing a wider range of reordering strategies.

We will first give an overview of the framework (§2). We then describe how to implement four reordering methods in a single decoder in §3. §4 presents some Chinese-English results on the NIST MT test sets. It also shows results on web log and broadcast news data.

## 2    Reordering in Decoding

### 2.1    Hypothesis

The process of MT decoding can be thought of as a process of hypothesizing target translations. Given an input source sentence of length L, the decoding is done segment by segment. A *segment* is simply an n-word source chunk, where $1 \leq n \leq L$. Decoding finishes when all source chunks are translated (some source words that have no target translations can be thought of as being translated into a special token NULL). The decoder at this point outputs its best hypothesis.

### 2.2    Hypothesis with reorderings

In order to facilitate various search strategies, a separation of duty is called for. The decoder is composed of two major modules, a reordering module and a production module. The reordering module decides which source segment to be translated next. The production module produces the actual translations for a given segment. Although most of the start-of-the-art decoders have these two modules, they are nevertheless tightly coupled. Here they are separated. This separation does not compromise the search space of the decoder. Hypotheses that are explored in the traditional way are still explored in this framework. This separation is essential if one were to design a decoder that incorporates phrase-based, syntax-based, and

other types of decoding in a unified and disciplined way. In the decoder, each hypothesis carries with it a sequence of source segments to be decoded at the current time step. After the production module translates these segments and after beam pruning is applied to all the hypotheses produced at this time step, the hypotheses go back to the reordering module which determines the next source segments to be translated. This process continues until all source words are translated.

One can think of the reordering module as a black box whose sole responsibility is to determine the next sequence of source segments to be translated. Given this separation, the reordering module can be implemented in whichever way and the changes in it do not require changes to any other modules in the decoder. There can be a suite of such modules, each exploring different features and implementing different search schemes. A reordering module that implement basic distance-based reordering will take two parameters, the number of source words to be skipped and the window size that determines when the skipped words must be translated. A reordering module that is based on HIERO rules will take the library of HIERO rules and select the subset that fire on a given input sentence. The module will use this subset of rules to determine the source translation order. A parse-inspired reordering module will take an input parse tree and based on either a trained model or hand-written rules decide the next source sequence to be translated. As long as all the reordering modules are written to a common interface, they can be separately written and maintained.

Figure 1 shows an example of how three reordering modules can be incorporated into a single decoder. The input source is $S_1 \ldots S_n$.
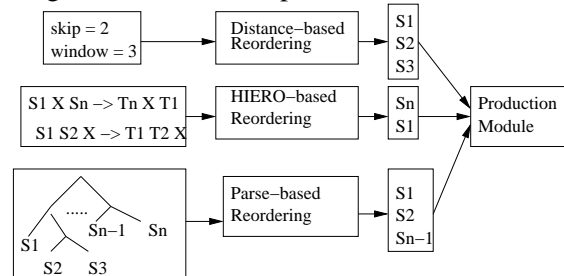


Figure 1.  Reordering module example

Each reordering module has its own resources and parameters which are shown on the left side. Each reordering module produces a vector of next source positions. The production module takes these positions and produces translations for them.

## 3 Reordering Modules

In this section, we describe four reordering modules implementing different reordering strategies. The framework is not limited to these four methods. We present these four to demonstrate the ability of the framework to incorporate a wide variety of reordering methods.

### 3.1 Distance-based Skip Reordering

This is the type of reordering first presented by (Brown et.al. 1993) and was briefly alluded to in the above *Introduction* section. This method is controlled by 2 parameters:

Skip = number of words whose translations are to be delayed. Let us call these words *skipped* words.

WindowWidth (ww) = maximum number of words allowed to be translated before translating the skipped words.

This reordering module outputs all the possible next source words to be translated according to these two parameters. For illustration purposes, let us use a bit vector $B$ to represent which source words have been translated. Thus those that have been translated have value 1 in the bit vector, and those un-translated have 0. As an example, let skip = 2 and ww = 3, and an input sentence of length = 10. Initially, all 10 entries of B are 0. At the first time step, only the following are possible next positions:

a) 1 0 0 0 0 0 0 0 0 0 : translate $1^{st}$ word
b) 0 1 0 0 0 0 0 0 0 0 : skip $1^{st}$ word
c) 0 0 1 0 0 0 0 0 0 0 : skip $1^{st}$ and $2^{nd}$ words

At the next time step, if we want to continue the path of c), we have these choices:
1) we can leave the first 2 words open and continue until we reach 3 words (because ww=3)
  c1) 0 0 1 1 0 0 0 0 0 0
  c2) 0 0 1 1 1 0 0 0 0 0

2) or we can go back and translate either of the first 2 skipped words:
  c3) 1 0 1 0 0 0 0 0 0 0
  c4) 0 1 1 0 0 0 0 0 0 0

It is clear that the search space easily blows up with large skip and window-width values. Therefore, a beam pruning step is performed after partial hypotheses are produced at every time step.

### 3.2 HIERO Hierarchical Reordering

In this section we show an example of how the Hiero decoding method (Chiang 2005) can be implemented as a reordering module in this framework. This is not meant to show that our MT decoder is a synchronous CFG parser. This is a conceptual demonstration of how the Hiero rules can be used in a reordering module to decide the source translation order and thus used in a traditional phrase-based decoder. This module uses the Hiero rules to determine the next source segment to be translated. The example is Chinese-English translation. Consider the following Chinese sentence (word position and English gloss are shown in parentheses):

澳大利亚(1.Australia) 是(2. is) 与(3. with) 北韩 (4. North Korea) 有(5. have) 邦交(6. diplomatic relation) 的(7. NULL) 少数(8. few) 国家(9. country) 之一(10. one of)

Suppose we have two following Hiero rules:
澳大利亚 X → Australia X     (1)
是 X 之一 → is one of X    (2)

The left-hand-side of Hiero rules are source phrases and the right-hand-side is their English translation and the Xs are the non-terminals whose extent is determined by the source input against which the rules are tested for matching. A rule fires if its left-hand-side matches certain segments of the input.

Given the above Chinese input and the two Hiero rules, the Hiero decoder as described in (Chiang 2005) will produce a partial hypothesis "*Australia is one of*" by firing the two rules during parsing (see Chiang 2005 for decoding details). We will show how to decode in the Hiero paradigm using the framework.

The reordering module first decides a source segment based on rule (1). Rule (1) generates a sequence of source segments in term of source ranges: <[1,1],[2,10]>. This means the source segment spanning range [1,1] (word 1, *澳大利亚 /Australia*) is to be translated first, and then the remaining segment spanning range [2,10] is to be translated next. This is exactly what rule (1) dictates where *澳大利亚* corresponds to source [1,1] in the reordering module's output and the *X* is [2,10]. The range [1,1], after being given to the production module, results in the production of a partial hypothesis where the target "*Australia*" is produced. The task now is to translate the next source range [2,10]. At this point, the reordering module generates another source segment according to rule (2) where the left-hand-side "*是 X 之一*" is matched against the input and three corresponding source ranges are found which are *[2,2]* (*是/is*), *[4,9]* (*X*), and *[10,10]* (*之 一/one of*). According to rule (2), this source sequence is to be translated in the order of *[2,2]* (*is*), *[10,10]* (*one of*), and then *[4,9]* (*X*). Therefore the output of the reordering module at this stage is *<[2,2],[10,10],[4,9]>*. This would then go on to be translated and results in a partial hypothesis to "*Australia is one of*". Thus "*Australia is one of*" is a partial production which covers source segments [1,1] [2,2] and [10,10] in that order. Note that the source segments decoded so far are not contiguous and this is the effect of long-range reordering imposed by rule (2). The next stage is <[4,9]> which is what the *X* in rule (2) corresponds to. From here onwards, other rules will fire and the decoding sequence these rules dictate will be realized by the reordering module in the form of source ranges. This process can also be viewed hierarchically in Figure 2.

In Figure 2 the ranges (the bracketed numbers) are source segments and the leaves are English productions. Initially we have the whole input sentence as one range [1,10]. According to rule (1), this initial range is refined to be <*[1,1],[2,10]*>, the 2nd level in Figure 2. The [2,10] is further refined by rule (2) to generate the 3rd level ranges *<[2,2],[10,10],[4,9]>* and the process goes on. Ranges that cannot be further refined go into the production module which
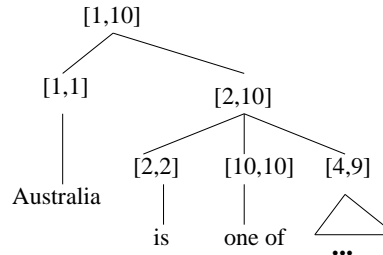


Figure 2. Hiero-style decoding

generates partial hypotheses which are the leaves in the figure. In other words, the partial hypotheses are generated by traversing the tree in Figure 2 in a left-to-right depth-first fashion.

## 3.3 Generalized Part-Of-Speech-based Reordering

The aim of a generalized part-of-speech-based reordering method is to tackle the problem of long-range word movement. Chinese is a pre-modification language in which the modifiers precede the head. The following is an example with English gloss in parentheses. The prepositional modifier "*on the table*" follows the head "*the book*" in English (3.3b), but precedes it in Chinese (3.3a). When the modifiers are long, word-based local reordering is inadequate to handle the movement.

3.3a. 桌(*table*) 上(*on*) 的(NULL) 书(*book*)
3.3b. the book on the table

There have been several approaches to the problem some of which are mentioned in §1. Compared to these methods, this approach is lightweight in that it requires only part-of-speech (POS) tagging on the source side. The idea is to capture general long-distance distortion phenomena by extracting reordering patterns using a mixture of words and part-of-speech tags on the source side. The reordering patterns are extracted for every contiguously aligned source segment in the following form:

*source sequence → target sequence*

Both the *source sequence* and the *target sequence* are expressed using a combination of source words and POS tags. The patterns are 'generalized' not only because POS tags are used but also because variables or place-holders are

allowed. Given a pair of source and target training sentences, their word alignments and POS tags on the source, we look for any contiguously aligned source segment and extract word reordering patterns around it. Figure 3 shows an example.

Shown in Figure 3 are a pair of Chinese and English sentence, the Chinese POS tags and the word alignment indicated by the lines. When multiple English words are aligned to a single Chinese word, they are grouped by a rectangle for easy viewing. Here we have a contiguously aligned source segment from position 3 to 8. Using the range notation, we say that source range [3,8] is aligned to target range [6, 14]. Let X denote the source segment [3,8]. The source verb phrase (at positions 9 and 10) occur after X whereas the corresponding target verb phrase (target words 2,3, and 4) occur before the translation of X (which is target [6,14]). We thus extract the following pattern:

$$对\ X\ V\ N \rightarrow V\ N\ 对\ X \qquad (1)$$

where the left-hand side '对 X V N' is the source word sequence and the right-hand side 'V N 对 X' is the target word sequence. The X in the pattern is meant to represent a variable, to be matched by a sequence of source words in the test data when this pattern fires during decoding. Note that the pattern is a mixture of words and POS tags. Specifically, the word identity of the preposition 对 (position 2) is retained whereas the content words (the verb and the noun) are substituted by their POS tags. This is because in general, for the reordering purpose the POS tags are good class representations for content words whereas different prepositions may have different word order patterns so that mapping them all to a single POS $P$ masks the difference. Examples of patterns are shown in Table 1.

In Chinese-English translation, the majority of the reorderings occur around verb modifiers (prepositions) and noun modifiers (usually around the Chinese part-of-speech DEG as in position 6). Therefore we choose to extract only these 2 kinds of patterns that involve a preposition and/or a DEG. In the example above, there are only 2 such patterns:

$$对\ X\ V\ N \rightarrow V\ N\ 对\ X \qquad (1)$$
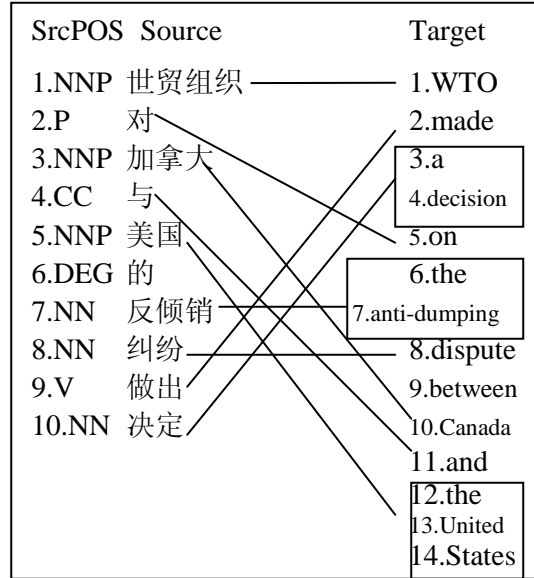$$X_1\ DEG\ X_2 \rightarrow X_2\ DEG\ X_1 \qquad (2)$$



Figure 3. Chinese/English Alignment Example

| | Source Seq. | Target Seq. | Count | P(tseq \|sseq) |
|---|---|---|---|---|
| 1 | X DEG NN | X DEG NN | 861 | 0.198 |
| 2 | X DEG NN | X NN DEG | 1322 | 0.305 |
| 3 | X DEG NN | NN DEG X | 2070 | 0.477 |
| 4 | X DEG NN | NN X DEG | 10 | 0.002 |
| 5 | X DEG NN | DEG NN X | 52 | 0.012 |
| 6 | X DEG NN | DEG X NN | 22 | 0.005 |
| 7 | 由 X VV | 由 X VV | 15 | 0.118 |
| 8 | 由 X VV | VV 由 X | 112 | 0.882 |
| 9 | 因为 X VV | VV 因为 X | 2 | 0.041 |
| 10 | 因为 X VV | 因为 X VV | 47 | 0.959 |

Table 1. Pattern examples

In the table, we see that when the preposition is 由 (rows 7 and 8, translation: *by*), then the swapping is more likely (0.882 in row 8). When the preposition is 因为 (rows 9 and 10 translation: *because*), then the target most often stays the same order as the source (prob 0.959, last row).

### 3.4 Parse-based Lexicalized Reordering

Part-of-speech reordering patterns as described in §3.3 are crude approximation to the structure of the source sentence. For example, in the source pattern 'X DEG NN', the variable X can match a source segment of arbitrary length which is followed by 'DEG NN'. Although it does capture very long range movement as a result of

X matching a long segment, it often searches unnecessarily for those segments that are implausible matches to X. The goal of the pattern 'X DEG NN' is to capture the pre-modification phenomenon in Chinese where X is to match a modifier. Parse trees are good at capturing these structures. A parse tree is shown in Figure 4a using notation from Chinese Treebank CHTB5 (nodes with same label are numbered for easy reference).

The node CP has 2 children, first of which is an IP and second is the word whose POS is DEG. This tree denotes a big NP (top node NP1) whose head is the rightmost NP (NP2). The IP under the CP is the modifier. Given this tree, we can easily tell the span of the modifier IP.



4a. NP rule          4b. VP rule

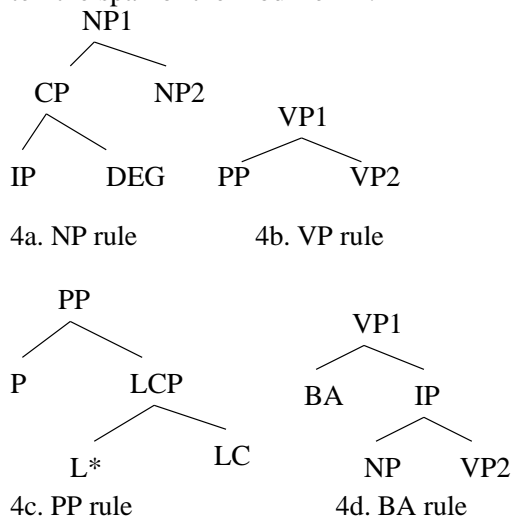4c. PP rule          4d. BA rule

Figure 4. Source parse trees to be reordered

Parse trees represent the whole structure of the entire sentence. Not every structure is of interest to the reordering problem. In a way similar to that used in part-of-speech-pattern extraction (§3.3), we restrict our attention to four kinds of structures, the first of which is NP involving a DEG (as in Figure 4a.) The other three are in Figure 4b, 4c, and 4d. In Figure 4c, the label L* means any node, sometimes it is a CP, sometimes an IP, and so on.

Figure 4b captures the pre-modification in case of a VP where PP modifies VP2 in Chinese and needs to be swapped when translating into English. Figure 4c is the case where there are both pre-position (P) and post-position (LC) in the Chinese. In English, there are only pre-positions and therefore something must be

done to the post-position LC. Figure 4d is the construction in Chinese that turns an SVO word order into SOV and here we want VP2 to precede its object NP.

The reordering rules are written using the leaves in the parse tree, in other words, the lexical items. In the rules below, we use the bracketed label [L] to mean the leaves it covers, so [NP] means the leaves under NP. The reordering rules for the 4 structures are:
NP (Figure 4a): [NP2] [DEG] [IP]
VP (Figure 4b): [VP2] [PP]
PP (Figure 4c): [P] [LC] [L*]
BA (Figure 4d): [VP2] [NP]
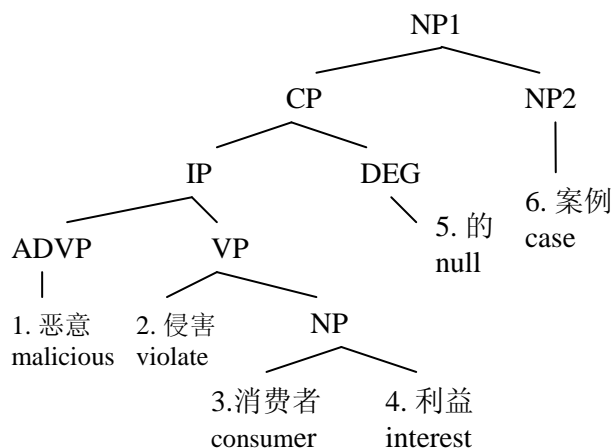
Figure 5 is an example of rule 4a.



Figure 5. Lexical example of NP rule

Chinese words and their English gloss are written at the leaves. The correct English translation is "*cases of malicious violation of consumer interests*". The DEG in the tree signals that the preceding IP is the modifier of the head NP2. Given this tree, the reordering rule is [NP2] [DEG] [IP] (see 4a) which will be written in the form

*source sequence → target sequence*

which is realized as the following (the indices are for easy reference and are not in the actual rule)
1.恶意 2.侵害病 3.消费者 4.利益 5.的 6.案例 → 6.案例 5.的 1.恶意 2.侵害病 3.消费者 4.利益

The first three of these structures are explored in (Wang et.al. 2007). The crucial difference is that in (Wang et.al. 2007), the reordering rules for

these structures are used as a hard decision to pre-order the source. Here the rules are used to extract reorder patterns which are used as an integral part of the decoder. The reordering module not only proposes the next source segment according to the reordering patterns but also proposes monotone choices. This is because first, the parser is errorful. In this work, we use the Stanford Parser (Levy and Manning 2003). On the last 929 sentences of CHTB5, the parser achieves 81% label F-measure on true CHTB5 word segmentation and drops to 65% on system segmentation using the Stanford CRF Segmenter (Tseng et.al. 2005). The second reason to let the decoder choose between reordering and monotone is other modules such as phrase tables and target LM can have an influence on the order choice too, especially when both reorder and monotone are acceptable as in the following example:

*CH: 我(my/mine/I/me) 的(DEG/null) 书(book)*
*English1: my book (monotone)*
*English2: the book of mine (reorder)*

Since the Chinese has a DEG, our reordering rule will prefer to swap but monotone is often correct . In cases like these we let the other models, such as TM and LM, to also have a say in deciding the outcome. The reordering module will present both choices to be produced.

## 4   Experiment Results

We run our experiments on NIST Chinese-English MT03 and MT04 and also on weblog (WL) and broadcast news (BN) data. The WL and BN test sets are held-out data from LDC-released parallel training data. WL data is from LDC2006E34 and BN is from LDC2006E10. The metric reported is cased *BLEUn4* 4-gram BLEU (Papineni et.al. 2001) .

We train HMM alignments in both direction to get source-to-target and target-to-source probabilities. We have a smoothed 5-gram English LM built on the English Gigaword corpus and the English side of the Chinese-English parallel corpora distributed by LDC from year 2000 to 2005.

For distance-based skip reordering (§3.1) we experimented with four sets of skip and WindowWidth values.

For part-of-speech reordering patterns, we use the 3259 hand alignments contained in LDC2006E93. We build a MaxEnt Chinese POS tagger and tagged the Chinese side of this data. The tagger achieves 92% F-measure on the 10% heldout data of CHTB5. We then extracted reordering patterns according to the procedure described in §3.3. A total of 788 source patterns were extracted. It is a small pattern set because of our specific extraction criteria described in §3.3. At decoding time, an average of 15-20 patterns fire on a single sentence. We use the unigram probabilities of the rules as shown in Table 1 to score the rules.

For parse-based lexical reordering rules, we run the Stanford parser on the test set and extract the lexicalized patterns. The number of patterns of each test set is shown in Table 2. The reordered rules are assigned a value of 0.9 and the monotones are assigned a value of 0.1.

| Test Set | # Sentences | # Lex.Patterns |
|---|---|---|
| MT03 | 919 | 4,824 |
| MT04 | 1,788 | 13,639 |
| WL (LDC2006E34) | 550 | 3,261 |
| BN (LDC2006E10) | 2,069 | 12,492 |

Table 2.  Test data statistics

The results on the NIST MT test sets MT03 and MT04 utilizing 4 references are in shown in Table 3. The results of the weblog and broadcast news data are shown in Table 4 where there is 1 reference for each set. The confidence intervals in these experiments are between ±0.l2 and ±0.16. This means the variations in rows 1-5 of Table 3 are not statistically significant. The part-of-speech based reordering shows marginal improvement. We see significant improvement in using parse-based reordering rules.

| | Cased-BLEU**r4**n4 | MT03 | MT04 |
|---|---|---|---|
| 1 | Skip0 (monotone) | 0.2817 | 0.3023 |
| 2 | Skip = 1; WW=2 | 0.2854 | 0.3024 |
| 3 | Skip = 2; WW = 3 | 0.2878 | 0.3061 |
| 4 | Skip = 3; WW = 4 | 0.2903 | 0.3081 |
| 5 | Skip = 4; WW = 5 | 0.2833 | 0.3090 |
| 6 | Generalized POS | 0.3066 | 0.3182 |
| 7 | Parse-based Lex | 0.3231 | 0.3250 |

Table 3. NIST MT03 and MT04 Results

| Cased-BLEU**r1**n4 | Weblog | Broadcast News |
|---|---|---|
| Skip0 (monotone) | 0.0656 | 0.0858 |
| Generalized POS | 0.0694 | 0.0878 |
| Parse-based Lex | 0.0862 | 0.1135 |

Table 4. Weblog and BN results

## 5. Conclusions

We have presented a decoding framework that greatly facilitates the incorporation of various reordering strategies that are necessary to put the words in the right order during translation. This modularized framework abstracts away the reordering phase from the rest of the decoder components. This not only makes the decoder easier to maintain but also allows rapid experimentation of a variety of reordering methods. Instead of using one reordering module, multiple reordering modules are used to come up with a list of next possible source segment choices. So far we have not seen any significant improvement using combination of reordering modules. This warrants further research since intuitively the knowledge-rich modules and the distance-based methods ought to complement each other. The POS and parse-based methods are very targeted and work quite well when the source structure is correctly understood, but cannot correct itself when errors occur in the tagging or the parsing process. The distance-based methods pay no attention to structure and is thus immune from source processing errors.

Although we present the POS-based and parse-based reordering modules in the context of Chinese to English translation, they can be used for other languages as well. For example, in Arabic to English translation, we extract patterns that capture the VSO word order of Arabic (English is SVO) and also the adjectival post-modification of noun.

The framework greatly reduces the amount of work needed to experiment with drastically different ways of reordering. All these can now be done in one single decoder.

## Acknowledgements

## References

P.F.Brown, S.A.Della Pietra, V.J.Della Pietra, and R.L.Mercer. *The Mathematics of Statistical Machine Translation.* Computation Linguistics, 19(2).

D. Chiang 2005 *A hierarchical phrase-based model for statistical machine translation.* 2005 ACL.

Y.Lee and N.Ge 2006 *Local reordering in statistical machine translation.* Workshop of TCStar 2006

R. Levy and C. Manning. 2003. *Is it harder to parse Chinese, or the Chinese Treebank?* ACL 2003

S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney. *Novel Reordering Approaches in Phrase-Based Statistical Machine Translation.* In ACL Workshop on Building and Using Parallel Texts 2005

F.Och, P. Koehn, and D. Marcu. 2003. *Statistical phrase-based translation.* HLT-NAACL 2003

F.Och, C.Tillmann, H.Ney 1999 *Improved alignment models for statistical machine ranslation*, EMNLP

F. Och. 2003. *Minimum error rate training in statistical machine translation.* ACL2003

K.Papineni, S.Roukos, T.Ward, W.Zhu 2001. *A method for automatic evaluation for MT,* ACL 2001

C.Tillmann, H. Ney 2000 *Word reordering and DP-based search in SMT*, COLING 2000

H. Tseng, P. Chang, G. Andrew, D. Jurafsky and C. Manning. *A Conditional Random Field Word Segmenter.* In Fourth SIGHAN Workshop 2005.

Dekai Wu. 1997 *Stochastic inversion transduction grammars and bilingual parsing of parallel corpora.* Computational Linguistics, Vol. 23, pp. 377-404

Kenji Yamada and Kevin Knight 2001 *A syntax-based statistical translation model.* ACL 2001

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

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