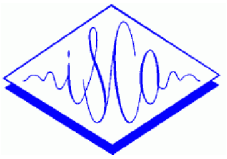
ISCA Archive
http://www.isca-speech.org/archive



International Workshop on Spoken
Language Translation (IWSLT) 2006
Keihanna Science City, Kyoto, Japan
November 27-28, 2006

Reordering Rules for Phrase-based Statistical Machine Translation

Boxing Chen, Mauro Cettolo, Marcello Federico

ITC-irst - Centro per la Ricerca Scientifica e Tecnologica
via Sommarive, 18
38050 Povo (Trento), Italy
{boxing,cettolo,federico}@itc.it

Abstract

This paper proposes the use of rules automatically extracted from word aligned training data to model word reordering phenomena in phrase-based statistical machine translation. Scores computed from matching rules are used as additional feature functions in the rescoring stage of the automatic translation process from various languages to English, in the ambit of a popular traveling domain task. Rules are defined either on Part-of-Speech or words. Part-of-Speech rules are extracted from and applied to Chinese, while lexicalized rules are extracted from and applied to Chinese, Japanese and Arabic.

Both Part-of-Speech and lexicalized rules yield an absolute improvement of the BLEU score of 0.4-0.9 points without affecting the NIST score, on the Chinese-to-English translation task. On other language pairs which differ a lot in the word order, the use of lexicalized rules allows to observe significant improvements as well.

1. Introduction

In Machine Translation (MT), one of the main problems to handle is word reordering. Informally, a word is "reordered" when it and its translation occupy different positions within the corresponding sentences.

In Statistical Machine Translation (SMT) [1], word reordering is faced from two points of view: constraints and modeling. If arbitrary word-reorderings are permitted, the exact decoding problem was shown to be NP-hard [2]; it can be made polynomial-time by introducing proper constraints, such as IBM constraints [3] and Inversion Transduction Grammars (ITG) constraints [4, 5]. It is worth to notice that both types of constraints are linguistically blind, i.e. they are unable to tune the number of allowed word reorderings according to the actual portion of the input sentence under process.

Whatever the constraint, among the allowed word-reorderings it is expected that some are more likely than others. The aim of reordering models, known also as distortion models, is just that of providing a measure of the plausibility of reorderings. Most of the distortion models developed so far are unable to exploit linguistic context to score reorderings: they just predict target positions on the basis of other (source and target) positions.

Some lexicalized block re-ordering models were presented in [6, 7, 8], where each block is associated with an orientation with respect to its predecessor. During decoding, the probability of a sequence of blocks with the corresponding orientations is computed. In [9] and [10], the aim is to capture particular syntactic phenomena occurring in the source language which are not preserved by the target language. Part-of-Speech (POS) rules are applied for preprocessing the source side both in translation model training and in decoding.

In this work we present a novel method for extracting reordering rules from word-aligned training data. The units in the left-hand-side of rules can be plain words or POS's; moreover, rules can reorder sequences of single units or a pair of unit blocks. In a two-stage SMT system like our one, reordering rules could be exploited directly during decoding in order to focus the search on reordering phenomena observed in training data. Here, we employed them in the rescoring stage, in terms of proper additional feature functions.

This paper is organized as follows. Section 2 presents the phrase-based SMT system we have worked with. Section 3 introduces the new reordering method. Sections 4 and 5 present the experimental results and future application, respectively. Some conclusions are drawn in Section 6.

2. The Phrase-based SMT System

Given a string $\mathbf{f}$ in the source language, the goal of statistical machine translation is to select the string $\mathbf{e}$ in the target language which maximizes the posterior distribution $\Pr(\mathbf{e} \mid \mathbf{f})$. In phrase-based translation, words are no longer the only units of translation, but they are complemented by strings of consecutive words, the phrases. By assuming a log-linear model [11, 12] and by introducing the concept of word alignment [1], the optimal translation can be searched for with the criterion:

$$\tilde{\mathbf{e}}^* = \arg\max_{\tilde{\mathbf{e}}} \max_{\mathbf{a}} \sum_{r=1}^{R} \lambda_r h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}),$$

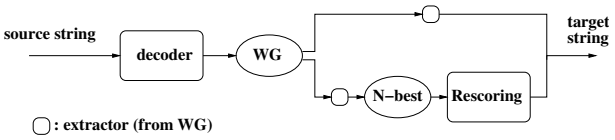


Figure 1: Architecture of the ITC-irst SMT system. The decoder produces a word-graph (WG) of translation hypotheses. In single-stage translation the most probable string is output. In two-stage decoding, N-best translations are extracted, re-scored, and re-ranked by applying additional feature functions.

where $\tilde{\mathbf{e}}$ represents a string of phrases in the target language, $\mathbf{a}$ an alignment from the words in $\mathbf{f}$ to the phrases in $\tilde{\mathbf{e}}$, and $h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a})$ $r = 1, \ldots, R$ are *feature functions*, designed to model different aspects of the translation process.

Figure 1 illustrates how the translation of an input string is performed by the ITC-irst SMT system. In the first stage, a beam search algorithm (decoder) computes a word graph of translation hypotheses. Hence, either the best translation hypothesis is directly extracted from the word graph and output, or an N-best list of translations is computed by means of an exact algorithm [13]. The N-best translations are then re-ranked by applying additional feature functions and the top ranking translation is finally output.

The search algorithm [14] exploits dynamic programming, i.e. the optimal solution is computed by expanding and recombining previously computed partial theories. The target string is extended step by step by covering new source positions until all of them are covered. For each added target phrase, a source phrase within the source string is chosen, and the corresponding score is computed on the basis of its position and phrase-to-phrase translation probabilities. The fluency of the added target phrase with respect to its left context is evaluated by a 4-gram language model. Some exceptions are also managed: target phrases might be added which do not translate any source word, and some of the source words can be left untranslated, that is they are translated with a special empty word.

To cope with the large number of generated theories, a beam is used to prune out partial theories that are less promising and constraints are set to possible word re-ordering. Word re-ordering constraints are applied during translation each time a new source position is covered, by limiting the number of vacant positions on the left and the distance from the left most vacant position.

The log-linear model on which both the search algorithm and the rescoring stage work embeds feature functions whose parameters are either estimated from data or empirically fixed. The scaling factors λ of the log-linear model are instead estimated on a development set, by applying a *minimum error training* procedure [15, 16].

The language model feature function is estimated on unsegmented monolingual texts.

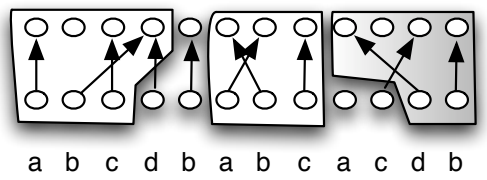


Figure 2: Example illustrating the concept of block.

The phrase-to-phrase probability feature is estimated from phrase-pair statistics extracted from word-aligned parallel texts. Alignments are computed with the GIZA++ software tool [17]. Phrase pairs are extracted from the segment pairs by means of the algorithm described in [18].

The distortion model feature function is a fixed negative exponential function computed on the distance between the current and the previously translated source phrases.

3. Reordering Rules

In order to overcome the limitations of our simple distortion model, we propose to enrich the translation process with reordering rules as defined in the following.

Units on which rules work can be words (lexicalized rules) or POS's (POS rules). Rules can suggest/constraint reorderings at the level of either single units or ngrams of units (blocks). In the following discussion, POS is the unit taken as reference; the extension to words is straightforward.

3.1. Definition of Block

A block is a sequence of source units all occurrences of which are aligned to consecutive positions in an aligned parallel corpus (null alignments are ignored).
In the example reported in Figure 2 the sequence "a b c" is a block, while the sequence "d b" is not, as its last occurrence does not satisfy the contiguity constraint on the target positions. Note that a single unit is also a block.

Blocks can be extracted at the level of words or POS by means of a word aligned parallel corpus. In the second case, the source side must be provided with a POS annotation.

3.2. Definition of Reordering Rule

A reordering rule consists of two sides: the left-hand-side (lhs), which is a POS pattern, and the right-hand-side (rhs), which corresponds to a possible reordering of that pattern. Different rules can share the lhs: in such cases, the same pattern can be reordered in more than one way. Rules are weighted, according to statistics extracted from training data.

There are two kinds of reordering patterns: unit-based, which define reorderings at the level of single POS's (unit reordering rules), and block-based, which define reorderings between whole blocks of POS's (block reordering rules). Let us consider the following examples:

- **Unit reordering rules:**
  - /rr /vmodal /v # 1 2 3 : 7 (18)
  - /rr /vmodal /v # 2 1 3 : 4 (18)
  - /rr /vmodal /v # 1 2 0 : 3 (18)
  - /rr /vmodal /v # 0 1 2 : 2 (18)
  - /rr /vmodal /v # 1 0 2 : 1 (18)
  - /rr /vmodal /v # 2 1 0 : 1 (18)
  - /v /d /v /m /q # 1 2 3 4 5 : 4 (5)
  - /v /d /v /m /q # 0 0 1 2 3 : 1 (5)
- **Block reordering rules:**
  - [/rr /vmodal /v] [/ng] # 1 2 : 3 (4)
  - [/rr /vmodal /v] [/ng] # 1 0 : 1 (4)

The tokens "/rr", "/vmodal", "/v", etc. are Chinese POS's, while the sequences "/rr /vmodal /v" and "/v /d /v /m /q" are POS patterns (p_1^n). The strings of numbers in between the symbols "#" and ":" represent suggested reordering (r_1^n): each integer r_i represents the new position of (the translation of) p_i. For example, the rhs of the second unit reordering rule is "2 1 3". The "2" in the first position means that "/rr" goes in the second position; the "1" in the second position means the "/vmodal" goes in the first position; finally, the "3" in the third position means that the "/v" keeps the third position. A "0" means that the corresponding token is deleted, i.e. it is aligned to the "null" word.

A pair of square brackets indicates a block. Block reordering ruless are always binary.

The two numbers after the colon (:) are collected from training data and are respectively the number of times the rhs (reordering suggestion) of the rule has been observed $count(r_1^n)$ and (inside brackets) the number of occurrences of the rule pattern $count(p_1^n)$. The probability of each reordering suggestion is computed as:

$$P(r_1^n|p_1^n) = \frac{count(r_1^n)}{count(p_1^n)} \quad (1)$$

3.3. Extraction of Blocks

Given the above definition, blocks are extracted from source-to-target (direct) aligned training data by means of the procedure shown in Figure 3. For each source sentence s in the training data, it is assumed that the direct alignment $\mathbf{a}$ is available. For each source n-gram (with n up to 20) its counter is updated (lines 2-10). Then, it is checked if the n-gram actually corresponds to a block: lines 12-18 check if there are source words on its left that are aligned to indexes within the n-gram translation; lines 19-25 check if there are source words on its right that are aligned to indexes within the n-gram translation. If both checks are passed, a second counter is updated (lines 26-28). Finally, only those n-grams which have been observed as blocks and more than once are returned as actual blocks (lines 29-34).

```
1  BLCK = NULL // Array of Blocks
2  for each source sentence s in training data
3      do
4          l = length(s)
5          for j from 1 to l
6              do
7                  for n from 0 to min{l-j,20}
8                      do
9                          ngram = f[j...j+n]
10                         freq(ngram)++
11                         isBLCKtag = 1
12                         k=1
13                         while isBLCKtag==1 && k <j
14                             do
15                                 if a(k) >min{a(j)...a(j+n)} &&
                                      a(k) <max{a(j)...a(j+n)}
16                                   then
17                                       isBLCKtag = 0
18                                 k++
19                         k=j+n+1
20                         while isBLCKtag==1 && k <= l
21                             do
22                                 if a(k) >min{a(j)...a(j+n)} &&
                                      a(k) <max{a(j)...a(j+n)}
23                                   then
24                                       isBLCKtag = 0
25                                 k++
26                         if isBLCKtag == 1
27                           then
28                                 blck_freq(ngram) ++
29 for each ngram
30     do
31         if blck_freq(ngram) == freq(ngram) && freq(ngram) >1
32           then
33                 push(BLCK, ngram)
34 return BLCK
```

Figure 3: Procedure for the extraction of blocks.

The check at line 31 guarantees that only real blocks are collected which occur more than once. Experimentally, this results in short blocks. By relaxing a bit the check, we can provide a larger set of "quasi-blocks" which are longer than the original ones. Hence, we replace the condition at line 31 with blck_freq(ngram)/freq(ngram)$\geq \theta$, where the ratio is compared to a threshold ($0 \leq \theta \leq 1$). The smaller θ is the more are the extracted quasi-blocks. For $\theta = 1$ the check is equivalent to the original one. Empirically, we found that with θ set to 0.9 a larger number of still reliable quasi-blocks is collected, which permits to model longer span reordering phenomena.

3.4. Extraction of Reordering Rules

If an n-gram is a block, then reordering rules for it can be easily extracted by looking at all the observed relative or intra-alignments of its words. The extraction is done by means of the procedure shown in Figure 4. It is assumed that the direct alignment $\mathbf{a}$ and the set of blocks BLCK, computed through the procedure described in the previous section, are available. For each source sentence, all possible ngrams are generated (lines 3-8). Only for ngrams which are blocks

```
1  RPU = NULL // Array of Unit Reordering Patterns
2  RPB = NULL // Array of Block Reordering Patterns
3  for each source sentence s in training data do
4      l = length(s)
5      for j from 1 to l
6      for n from 0 to min{l-j,20}
7        do
8          ngram = f[j...j+n]
9          if ngram ∈ BLCK
10             then
11               freq(ngram)++
12               order = a[j...j+n]
13               UnitReorderRule = ngram+#+order
14               freq(UnitReorderRule)++
15               push(RPU,UnitReorderRule)
16               for k from j+n+1 to l
17               for m from k+1 to l do
18                  right = f[k...m]
19                  if right ∈ BLCK then
20                   if max{a(j)...a(j+n)} == 0 && max{a(k)...a(m)}==0
21                     then BlockReorderSuggestion = "0 0"
22                   elseif max{a(j)...a(j+n)} == 0 && max{a(k)...a(m)} >0
23                     then BlockReorderSuggestion = "0 1"
24                   elseif max{a(j)...a(j+n)} >0 && max{a(k)...a(m)} == 0
25                     then BlockReorderSuggestion = "1 0"
26                   elseif min{a(j)...a(j+n)} >= max{a(k)...a(m)}
27                     then BlockReorderSuggestion = "2 1"
28                   elseif max{a(j)...a(j+n)} <= min{a(k)...a(m)}
29                     then BlockReorderSuggestion = "1 2"
30                   else BlockReorderSuggestion = "mixed"
31                   if k == j+n+1
32                     then BlockReorderPattern = ngram+right
33                     else  BlockReorderPattern = ngram+"HOLE"+right
34                   BlockReorderRule =
35                      BlockReorderPattern+#+BlockReorderSuggestion
36                   freq(BlockReorderPattern)++
37                   freq(BlockReorderRule)++
38                   push(RPB,BlockReorderRule)
39 return RPU, RPB, freq
```

Figure 4: Procedure for the extraction of reordering rules.

(check in line 9), the corresponding intra-alignment is stored as possible unit reordering rule (lines 11-15). Block reordering rules are generated according to the relative position of the alignments of the current block and each right-hand block (lines 16-30). Different rules are defined if the two blocks result to be consecutive or not (lines 31-33). Storing of rules is performed in lines 34-38. Finally, the procedure outputs the set of unit and block reordering rules, together with the corresponding counters (line 39).

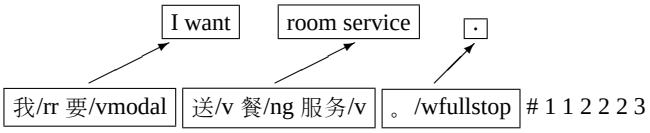
Actually, the comparison of alignments of current and right-hand blocks in lines 20, 22, 24, 26 and 28 could be replaced by a simpler check, e.g. the comparison of the last alignment of the first block with the first alignment of the second block. We need a stronger check since BLCK could contain quasi-blocks, as explained at the end of the previous section. This is the reason for which it is possible to enter in the branch "mixed" at line 30.

3.5. Use of Reordering Rules

Once reordering rules are available, they can be applied to each input sentence. The possibility we have investigated in this work is to use them in the rescoring stage as additional feature function.

Table 1 shows an input sentence and a set of reordering rules matching some portions of it. The symbol "*" stands for ignored positions.

In each entry of the N-best translations used in the rescoring stage, the information about source and target phrases and the corresponding alignment are kept. As an example, let us consider the following N-best entry:



The first two Chinese words 我/rr 要/vmodal are translated by a single phrase ("I want") which is put in the first position of the target string; the words in positions 3-5: 送/v 餐/ng 服务/v are translated by a single phrase ("room service") which is put in position 2; the translation of punctuation 。/wfullstop is put in position 3 of the target string.

In the rescoring stage, we used unit reordering rules and block reordering rules as two different feature functions.

Before detailing how matching reordering rules can be used as an additional feature function, we have to handle an inconsistency which regards unit reordering rule. The problem derives from the fact that the decoder generates translations at the level of phrases rather than words: in fact, target reordering indexes of the above reported example (1, 2 and 3) refer to phrases, not to single words, i.e. the word order inside the source and target phrases is not provided. On the contrary, unit reordering rules involve the position of single units. To fill this gap, we used a flexible matching strategy: any rule which is compatible with the N-best list entry is accepted.

In the above example, the words 我/rr and 要/vmodal are translated by a unique target phrase put in position 1. Since no information is available on the exact order of the translation of the two words, any rule whose lhs matches the source of the entry and reorders the two words with no insertions ("1 2" or "2 1") or delete one of them ("1 0" or "0 1") is activated.

Now that we know how to match rules and N-best entries, it is possible to select, among all the rules whose lhs (p_1^n) matches fully or partially the source string, those with the rhs (r_1^n) matching the actual target phrase order. The corresponding probabilities $P(r_1^n|p_1^n)$ are taken and combined as follows:

$$h_{\texttt{rules}}(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}) = \frac{1}{K}(\sum_{i=1}^{K} \log P(r_{(i)1}^{\ n}|p_{(i)1}^{\ n})) \qquad (2)$$

Table 1: Example of input sentence and list of reordering rules.

	Test sentence:	我/rr	要/vmodal	送/v	餐/ng	服务/v	。/wfullstop	
a)	UnitREORDER	/rr	/vmodal	*	*	*	*	# 2 1 : 1047 (2356)
b)	UnitREORDER	/rr	/vmodal	*	*	*	*	# 1 2 : 1203 (2356)
c)	UnitREORDER	*	*	*	/ng	/v	*	# 1 2 : 68 (104)
d)	UnitREORDER	*	*	*	/ng	/v	*	# 1 0 : 13 (104)
e)	BlockREORDER	[/rr	/vmodal]	[/v]	*	*	*	# 1 2 : 1219 (1283)
f)	BlockREORDER	[/rr	/vmodal]	[/v]	*	*	*	# 1 0 : 43 (1283)
g)	BlockREORDER	[/rr	/vmodal]	*	[/ng	/v]	*	# 1 2 : 13 (16)
h)	BlockREORDER	[/rr	/vmodal]	*	[/ng	/v]	*	# 2 1 : 3 (16)
i)	BlockREORDER	*	*	[/v]	[/ng	/v]	*	# 2 1 : 2 (18)
j)	BlockREORDER	*	*	[/v]	[/ng	/v]	*	# 1 2 : 11 (18)
k)	BlockREORDER	*	*	*	[/ng	/v]	[/wfullstop]	# 1 2 : 14 (14)

where K is the number of the reordering patterns matching the given source/target pair. If for a matching POS pattern there is no reordering suggestion matching the actual translation, a small probability is used in the sum (2).

4. Experiments

4.1. Translation Tasks and Data

Experiments were carried out on the Basic Traveling Expression Corpus (BTEC) [19]. BTEC is a multilingual speech corpus that contains translation pairs taken from phrase books for tourists. We conducted experiments on three language pairs: Chinese-to-English, Japanese-to-English and Arabic-to-English. On the Chinese-to-English direction, both POS and lexicalized rules were tested. On the contrary, for the other three language pairs, only lexicalized rules were experimented: lexicalized rules were extracted and applied in the same way as POS rules, with the only obvious difference that here the lhs is defined on words instead of POS's. Detailed statistics on BTEC training data are reported in Table 2.

Data sets distributed for the CSTAR 2003 Evaluation Campaign were employed for development, while testing was performed on sets of IWSLT 2004 and IWSLT 2005 Evaluation Campaigns, and on one of the development sets (devset4) distributed for the IWSLT 2006 Evaluation Campaign. For each source sentence of those sets, 16 or 7 references are available. Detailed statistics are reported in Table 3.

The Chinese word segmentation and POS tagging were performed by means of ICTCLAS [20] for the experiments involving POS rules. In the experiments using lexicalized rules, Chinese and Japanese were re-segmented with an in-house word segmentation tool. We found that this permits to smooth inconsistencies of the manual segmentation. All texts were finally tokenized and put in lower case.

In the following, translation performance are provided in terms of BLEU [21] and NIST[1] scores, computed in the case insensitive modality and taking into account punctuation marks.

[1] http://www.nist.gov/speech/tests/mt/

4.2. Experimental Results

The set-up of the search algorithm was similar for all language pairs. Constraints on word reordering (cf. Section 2) are defined by means of the maximum vacancy number (MVN) and maximum vacancy distance (MVD) parameters. Preliminary investigations suggested the following settings:

Chinese-to-English:	MVD=6	MVN=6
Japanese-to-English:	MVD=8	MVN=8
Arabic-to-English:	MVD=4	MVN=4

For each input sentence, at most 1000 translation candidates were extracted. Actually, very few source sentences had a thousand of possible translations. This is due to both the shortness of source sentences and the limited amount of training data which yields the estimation of small size models.

Table 6: Translation examples for the Chinese-to-English task with POS rules.

baseline	can i try on this sweater cotton ?
rescored	can i try on this cotton sweater ?
reference	may i try on this cotton sweater ?
baseline	are there any clubs and pick up service rental ?
rescored	do you have any rental clubs and pick up service ?
reference	do you have rental clubs or a pick up service ?
baseline	i can get where a city map ?
rescored	where can i get a city map ?
reference	where can i get a map of the city ?

Translation performance on development and test sets are reported in Tables 4 and 5. Rows "baseline" provide scores of the 1-best translation, that is the decoder performance. The other rows show scores after the rescoring stage using as additional feature the unit reordering rules (rows "unit"), the block reordering rules (rows "block") and the two features

Table 2: Statistics of training data.

	Chi-POS	Chinese	Japanese	English	Arabic	English
Sentences	39,953				19,972	
Running words	359K	347K	392K	363K	180K	182K
Vocabulary	10,598	11,439	12,667	9,938	15,888	7,326

Table 3: Statistics of development and testing data.

		Chi-POS	Chinese	Japanese	Arabic	English
Dev	Sentences	506				506×16
CSTAR03	Running words	3,529	3,469	4,079	3,537	65,622
Test1	Sentences	500				500×16
IWSLT04	Running words	3,593	3,553	4,046	3,639	64,897
Test2	Sentences	506				506×16
IWSLT05	Running words	3,839	3,789	4,141	3,673	66,286
Test3	Sentences	489				489×7
Devset4	Running words	5,137	5,094	5,840	5,108	39,386

together (rows "unit+block"). The rescoring weights were estimated on the development sets through the minimum error training procedure mentioned in Section 2.

The first column quantifies the behavior of the POS rules for the Chinese-to-English task. In terms of BLEU score, block reordering rules outperform unit reordering rules; the use of the two features together results in a further benefit, allowing a global absolute improvement of 0.5-0.9 %BLEU on the three test sets (48.63% to 49.16%, 55.58% to 56.04%, and 16.45% to 17.36%).

The other columns report performance of lexicalized rules on the three considered language pairs. Focusing on the Chinese-to-English task, it seems to emerge that POS rules and lexicalized rules are equally effective. Again, improvements are obtained by using together block and unit reordering rules on language pairs with very different word order, such as Japanese-to-English and Arabic-to-English.

It is worth noticing that the improvements in BLEU score are obtained without penalizing the NIST score.

Some translation examples from the Chinese-to English task with POS rules are shown in Table 6.

5. Future Application

One possible extension of our approach is the generation of rules which span the whole input. This could be obtained by combining the partial block and word reordering patterns. As an example, consider again the sentence in Table 1.

First, we could fully cover the source sentence at the level of blocks. In fact, by combining rules e)-k), patterns covering all input positions are shown in Table 7, with the corresponding block reordering probabilities. Then, each block could be reordered internally at the level of words/POS's by exploiting unit reordering rules: Table 8 gives such reorderings, and the corresponding probabilities, obtained by combining rules a)-d) for the first block reordering rule of Table 7.

Table 8: Examples of unit reordering rules covering the whole input.

Pattern: /rr /vmodal /v /ng /v /wfullstop
unit reordering: 1 2 3 4 5 6 p=p_1 × 1047/2356 × 68/104
unit reordering: 2 1 3 4 5 6 p=p_1 × 1203/2356 × 68/104
unit reordering: 1 2 3 4 0 5 p=p_1 × 1047/2356 × 13/104
unit reordering: 2 1 3 4 0 5 p=p_1 × 1203/2356 × 13/104

Finally, let us consider the Chinese-to-English example of Table 9. The table shows the POS rule (*rule*) extracted from the training data that matches one sentence selected from the test set whose words (*source wrd*), POS's (*source pos*) and word-by-word translation (*wrd-by-wrd*) are displayed.

The sentence is a standard wh- question. In English the adverb is placed at the beginning of the sentence, while in Chinese it occurs in a position which depends on its function. In the example, it is placed between the subject ("bus") and the verb ("leave"). A sentence with different words but exactly the same structure was observed in the training data (*source* and *target wrd train*): from it, the reordering rule which matches the sentence allows to re-arrange the words in such a way that the order of the reference (first row) can be obtained. Note that the current decoder is unable to recover the right order (*mt out*). In fact, since MVD and MVN are set to 6, the English word "when" cannot be put by the decoder in the first target position as it is the translation of the source words in position 8 and 9. On the contrary, the reordering rules would make it possible. Their integration into the search process would allow the decoder to cover complex word reordering phenomena, keeping under control the size of the search space.

Table 4: BLEU% and NIST scores on development sets CSTAR03.

	Chi-POS		Chinese		Japanese		Arabic	
	BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
baseline	45.08	6.587	45.57	6.650	52.09	8.143	59.34	8.831
unit	45.93	6.507	45.83	6.568	52.82	8.152	59.54	8.869
block	46.00	6.573	46.89	6.578	53.06	8.197	59.61	8.823
unit+block	46.06	6.576	46.21	6.717	53.16	8.226	59.63	8.865

Table 5: BLEU% and NIST scores on test sets IWSLT 2004 and 2005.

		Chi-POS		Chinese		Japanese		Arabic	
		BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
IWSLT 2004	baseline	48.63	7.387	48.79	7.555	48.88	8.031	55.31	9.031
	unit	49.04	7.333	49.34	7.538	49.15	8.052	55.50	9.028
	block	49.10	7.356	49.34	7.572	49.35	8.013	55.55	9.049
	unit+block	49.16	7.406	49.42	7.562	49.41	8.049	55.63	9.050
IWSLT 2005	baseline	55.58	8.493	57.30	8.732	50.65	8.301	52.73	8.825
	unit	55.65	8.472	57.66	8.700	50.84	8.224	52.91	8.807
	block	55.77	8.488	57.76	8.722	50.92	8.232	53.15	8.815
	unit+block	56.04	8.492	57.82	8.726	51.32	8.260	53.35	8.831
Devset4	baseline	16.45	6.026	17.05	6.175	16.24	5.958	21.24	5.737
	unit	16.99	6.065	17.27	6.174	16.46	6.029	21.57	5.743
	block	16.93	6.047	17.19	6.198	16.34	5.983	21.31	5.694
	unit+block	17.36	6.069	17.44	6.203	16.61	6.011	21.64	5.739

6. Conclusions

We have presented a novel method to extract and use rules covering even complex word reordering phenomena occurring in statistical machine translation. Rules can be defined at the level of words or POS's; moreover, they can move single units or a pair of blocks of units.

They have been employed as additional feature functions in the rescoring stage of a statistical machine translation system, but they could also be integrated into the decoder.

Experiments were performed on the BTEC corpus and on three language directions: Chinese-to-English, Japanese-to-English and Arabic-to-English. Results showed that the reordering rules yield significant performance improvements for considered language pairs, which differ a lot in the word order.

7. Acknowledgements

This work has been funded by the European Union under the integrated project TC-STAR - Technology and Corpora for Speech-to-Speech Translation - (IST-2002-FP6-506738, http://www.tc-star.org).

8. References

[1] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, vol. 19(2), pp. 263–312, 1993.

[2] K. Knight, "Decoding complexity in word replacement translation models," *Computational Linguistics*, vol. 25(4), pp. 607–615, 1999.

[3] A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, A. S. Kehler, and R. L. Mercer, "Language translation apparatus and methods using context-based translation models," 1996, US Patent 5,510,981.

[4] D. Wu, "Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora," in *Proc. of HLT-EMNLP*, Montreal, Canada, 1995, pp. 1328–1335.

[5] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational Linguistics*, vol. 23(3), pp. 377–404, 1997.

[6] F. J. Och, D. Gildea, S. Khudanpur, and A. Sarkar, "A Smorgasbord of Features for Statistical Machine Translation," in *Proc. of NAACL*, Boston, MA, 2004.

[7] C. Tillmann, "A Unigram Orientation Model for Statistical Machine Translation," in *Companion Volume of the NAACL Proc.*, Boston, MA, 2004.

[8] C. Tillmann and T. Zhang, "A Localized Prediction Model for Statistical Machine Translation," in *Proc. of ACL*, Ann Arbor, Michigan, 2005, pp. 557–564.

Table 7: Examples of block reordering rules covering the whole input.

Pattern: [/rr /vmodal] [/v] [/ng /v] [/wfullstop]
block reordering: 1 2 3 4 {rules: e), g), j), k)} p_1 =1219/1283 × 13/16 × 11/18 × 14/14
block reordering: 1 0 2 3 {rules: f), g), j), k)} p_2 = 43/1283 × 13/16 × 11/18 × 14/14
block reordering: 1 3 2 4 {rules: e), g), i), k)} p_3 =1219/1283 × 13/16 × 2/18 × 14/14
block reordering: 2 3 1 4 {rules: e), h), i), k)} p_4 =1219/1283 × 3/16 × 2/18 × 14/14

Table 9: Example of POS reordering rule matching a full sentence.

mt out	at baltimore of the next bus depart from ?									
reference	when will the next bus to baltimore leave ?									
source wrd	到	巴尔的摩	的	下 一	班	巴士	什么 时候	开	？	
source pos	/p	/ns	/ude1	/f /m	/q	/n	/rq /rq	/v	/wquestion	
wrd-by-wrd	to	baltimore	∗	next	∗	bus	when	leave	?	
rule	/p	/ns	/ude1	/f /m	/q	/n	/rq /rq	/v	/wquestion	# 6 7 0 3 4 0 5 1 2 8 9
source wrd train	到	费城	的	下 一	列	火车	什么 时候	出发	？	
target wrd train	what time will the next train to philadelphia leave ?									

[9] Y.-S. Lee and S. Roukos, "IBM Spoken Language Translation System Evaluation," in *Proc. of IWSLT*, Kyoto, Japan, Sept. 2004, pp. 39–46, http://www.slt.atr.jp/IWSLT2004/archives/000619.html.

[10] Y.-S. Lee, "Morpho-Syntax in Statistical Machine Translation," in *OpenLab*, Trento, Italy, 2006, http://tc-star.itc.it/openlab2006/.

[11] A. Berger, S. A. D. Pietra, V. J. D. Pietra, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, vol. 22(1), 39-71, 1996.

[12] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. of ACL*, PA, Philadelphia, USA, 2002.

[13] B. Tran, F. Seide, and V. Steinbiss, "A Word Graph based N-Best Search in Continuous Speech Recognition," in *Proc. of ICLSP*, 1996.

[14] M. Federico and N. Bertoldi, "A word-to-phrase statistical translation model," *ACM Transaction on Speech Language Processing*, vol. 2(2), pp. 1–24, 2005.

[15] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of ACL*, Sapporo, Japan, 2003, pp. 160–167.

[16] M. Cettolo and M. Federico, "Minimum Error Training of Log-Linear Translation Models," in *Proc. of IWSLT*, Kyoto, Japan, Sept. 2004, pp. 103–106, http://www.slt.atr.jp/IWSLT2004/archives/000619.html.

[17] F. J. Och and H. Ney, "Improved Statistical Alignment Models," in *Proc. of ACL*, Hong Kong, China, 2000.

[18] B. Chen, R. Cattoni, N. Bertoldi, M. Cettolo, and M. Federico, "The ITC-irst SMT System for IWSLT-2005," in *Proc. of IWSLT*, 2005, http://www.is.cs.cmu.edu/iwslt2005/proceedings.html.

[19] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a Broad-Coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World," in *Proc. of 3rd LREC*, Las Palmas, Spain, 2002, pp. 147–152.

[20] H.-P. Zhang, Q. Liu, X. Q. Cheng, H. Zhang, and H.-K. Yu, "Chinese Lexical Analysis Using Hierarchical Hidden Markov Model," in *Proc. of SIGHAN Workshop on Chinese Language Processing*. Sapporo, Japan, 2003.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," IBM Research Division, Thomas J. Watson Research Center, Research Report RC22176, 2001.