

Reordering Constraints for Phrase-Based Statistical Machine Translation

Richard Zens¹, Hermann Ney¹, Taro Watanabe² and Eiichiro Sumita²

¹Lehrstuhl für Informatik VI
Computer Science Department
RWTH Aachen University, Germany
{zens,ney}@cs.rwth-aachen.de

²Spoken Language Translation Research Laboratories
ATR
Kyoto, Japan
{watanabe,sumita}@slt.atr.co.jp

Abstract

In statistical machine translation, the generation of a translation hypothesis is computationally expensive. If arbitrary reorderings are permitted, the search problem is NP-hard. On the other hand, if we restrict the possible reorderings in an appropriate way, we obtain a polynomial-time search algorithm. We investigate different reordering constraints for phrase-based statistical machine translation, namely the IBM constraints and the ITG constraints. We present efficient dynamic programming algorithms for both constraints. We evaluate the constraints with respect to translation quality on two Japanese–English tasks. We show that the reordering constraints improve translation quality compared to an unconstrained search that permits arbitrary phrase reorderings. The ITG constraints perform best on both tasks and yield statistically significant improvements compared to the unconstrained search.

1 Introduction

In statistical machine translation, we are given a source language (‘French’) sentence $f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language (‘English’) sentence $e_1^I = e_1 \dots e_i \dots e_I$. Among all possible target language sentences, we will choose the sentence with the highest probability:

$$\begin{aligned} \hat{e}_1^I &= \operatorname{argmax}_{e_1^I} \{Pr(e_1^I | f_1^J)\} \\ &= \operatorname{argmax}_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)\} \end{aligned}$$

This decomposition into two knowledge sources is known as the source-channel approach to statistical machine translation (Brown et al., 1990). It allows an independent modeling of target language model $Pr(e_1^I)$ and translation model $Pr(f_1^J | e_1^I)$. The target language model describes the well-formedness of

the target language sentence. The translation model links the source language sentence to the target language sentence. It can be further decomposed into alignment and lexicon model. The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. We have to maximize over all possible target language sentences.

An alternative to the classical source-channel approach is the direct modeling of the posterior probability $Pr(e_1^I | f_1^J)$. Using a log-linear model (Och and Ney, 2002), we obtain:

$$Pr(e_1^I | f_1^J) = \exp \left(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \right) \cdot Z(f_1^J)$$

Here, $Z(f_1^J)$ denotes the appropriate normalization constant. As a decision rule, we obtain:

$$\hat{e}_1^I = \operatorname{argmax}_{e_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \right\}$$

This approach is a generalization of the source-channel approach. It has the advantage that additional models or feature functions can be easily integrated into the overall system. The model scaling factors λ_1^M are trained according to the maximum entropy principle, e.g. using the GIS algorithm. Alternatively, one can train them with respect to the final translation quality measured by some error criterion (Och, 2003).

In this paper, we will investigate the reordering problem for phrase-based translation approaches. As the word order in source and target language may differ, the search algorithm has to allow certain reorderings. If arbitrary reorderings are allowed, the search problem is NP-hard (Knight, 1999). To obtain an efficient search algorithm, we can either restrict the possible reorderings or we have to use an approximation algorithm. Note that in

the latter case we cannot guarantee to find an optimal solution.

The remaining part of this work is structured as follows: in the next section, we will review the baseline translation system, namely the alignment template approach. Afterward, we will describe different reordering constraints. We will begin with the IBM constraints for phrase-based translation. Then, we will describe constraints based on *inversion transduction grammars* (ITG). In the following, we will call these the ITG constraints. In Section 4, we will present results for two Japanese–English translation tasks.

2 Alignment Template Approach

In this section, we give a brief description of the translation system, namely the alignment template approach. The key elements of this translation approach (Och et al., 1999) are the *alignment templates*. These are pairs of source and target language phrases with an alignment within the phrases. The alignment templates are built at the level of word classes. This improves the generalization capability of the alignment templates.

We use maximum entropy to train the model scaling factors (Och and Ney, 2002). As feature functions we use a phrase translation model as well as a word translation model. Additionally, we use two language model feature functions: a word-based trigram model and a class-based five-gram model. Furthermore, we use two heuristics, namely the word penalty and the alignment template penalty. To model the alignment template reorderings, we use a feature function that penalizes reorderings linear in the jump width.

A dynamic programming beam search algorithm is used to generate the translation hypothesis with maximum probability. This search algorithm allows for arbitrary reorderings at the level of alignment templates. Within the alignment templates, the reordering is learned in training and kept fix during the search process. There are no constraints on the reorderings within the alignment templates.

This is only a brief description of the alignment template approach. For further details, see (Och et al., 1999; Och and Ney, 2002).

3 Reordering Constraints

Although unconstrained reordering looks perfect from a theoretical point of view, we find that in practice constrained reordering shows

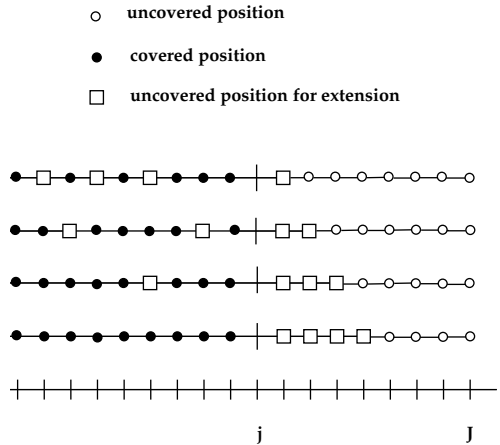


Figure 1: Illustration of the IBM constraints with $k = 3$, i.e. up to three positions may be skipped.

better performance. The possible advantages of reordering constraints are:

1. The search problem is simplified. As a result there are fewer search errors.
2. Unconstrained reordering is only helpful if we are able to estimate the reordering probabilities reliably, which is unfortunately not the case.

In this section, we will describe two variants of reordering constraints. The first constraints are based on the IBM constraints for single-word based translation models. The second constraints are based on ITGs. In the following, we will use the term “phrase” to mean either a sequence of words or a sequence of word classes as used in the alignment templates.

3.1 IBM Constraints

In this section, we describe restrictions on the phrase reordering in spirit of the IBM constraints (Berger et al., 1996).

First, we briefly review the IBM constraints at the word level. The target sentence is produced word by word. We keep a coverage vector to mark the already translated (covered) source positions. The next target word has to be the translation of one of the first k uncovered, i.e. not translated, source positions. The IBM constraints are illustrated in Figure 1. For further details see e.g. (Tillmann and Ney, 2003).

For the phrase-based translation approach, we use the same idea. The target sentence is produced phrase by phrase. Now, we allow skipping of up to k phrases. If we set $k = 0$, we obtain a search that is monotone at the phrase level as a special case.

The search problem can be solved using dynamic programming. We define an auxiliary function $Q(j, S, e)$. Here, the source position j is the first unprocessed source position; with unprocessed, we mean this source position is neither translated nor skipped. We use the set $S = \{(j_n, l_n) | n = 1, \dots, N\}$ to keep track of the skipped source phrases with lengths l_n and starting positions j_n . We show the formulae for a bigram language model and use the target language word e to keep track of the language model history. The symbol $\$$ is used to mark the sentence start and the sentence end. The extension to higher-order n -gram language models is straightforward. We use M to denote the maximum phrase length in the source language. We obtain the following dynamic programming equations:

$$\begin{aligned}
Q(1, \emptyset, \$) &= 1 \\
Q(j, S, e) &= \max \left\{ \begin{aligned} &\max_{e', \tilde{e}} \left\{ \max_{j-M \leq j' < j} Q(j', S, e') \cdot p(f_{j'}^{j-1} | \tilde{e}) \cdot p(\tilde{e} | e'), \right. \\ &\max_{\substack{(j', l) \in S' \\ S = S' \setminus \{(j', l)\}}} Q(j, S', e') \cdot p(f_{j'}^{j'+l-1} | \tilde{e}) \cdot p(\tilde{e} | e') \left. \right\}, \\ &\max_{\substack{j-M \leq j' < j \\ S': S = S' \cup \{(j', j-j')\} \wedge |S'| < k}} Q(j', S', e) \end{aligned} \right\} \\
Q(J+2, \emptyset, \$) &= \max_e Q(J+1, \emptyset, e) \cdot p(\$ | e)
\end{aligned}$$

In the recursion step, we have distinguished three cases: in the first case, we translate the next source phrase. This is the same expansion that is done in monotone search. In the second case, we translate a previously skipped phrase and in the third case we skip a source phrase. For notational convenience, we have omitted one constraint in the preceding equations: the final word of the target phrase \tilde{e} is the new language model state e (using a bigram language model).

Now, we analyze the complexity of this algorithm. Let E denote the vocabulary size of the target language and let \tilde{E} denote the maximum number of phrase translation candidates for a given source phrase. Then, $J \cdot (J \cdot M)^k \cdot E$ is an upper bound for the size of the Q -table. Once we have fixed a specific element of this table, the maximization steps can be done in $O(E \cdot \tilde{E} \cdot (M + k - 1) + (k - 1))$. Therefore, the complexity of this algorithm is in $O(J \cdot (J \cdot M)^k \cdot E \cdot (E \cdot \tilde{E} \cdot (M + k - 1) + (k - 1)))$. Assuming $k < M$, this can be simplified to $O((J \cdot M)^{k+1} \cdot E^2 \cdot \tilde{E})$. As already mentioned,

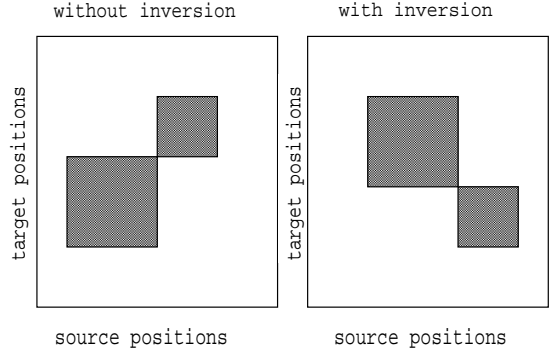


Figure 2: Illustration of monotone and inverted concatenation of two consecutive blocks.

setting $k = 0$ results in a search algorithm that is monotone at the phrase level.

3.2 ITG Constraints

In this section, we describe the ITG constraints (Wu, 1995; Wu, 1997). Here, we interpret the input sentence as a sequence of blocks. In the beginning, each alignment template is a block of its own. Then, the reordering process can be interpreted as follows: we select two consecutive blocks and merge them to a single block by choosing between two options: either keep the target phrases in monotone order or invert the order. This idea is illustrated in Figure 2. The dark boxes represent the two blocks to be merged. Once two blocks are merged, they are treated as a single block and they can be only merged further as a whole. It is not allowed to merge one of the subblocks again.

3.2.1 Dynamic Programming Algorithm

The ITG constraints allow for a polynomial-time search algorithm. It is based on the following dynamic programming recursion equations. During the search a table Q_{j_l, j_r, e_b, e_t} is constructed. Here, Q_{j_l, j_r, e_b, e_t} denotes the probability of the best hypothesis translating the source words from position j_l (left) to position j_r (right) which begins with the target language word e_b (bottom) and ends with the word e_t (top). This is illustrated in Figure 3.

The initialization is done with the phrase-based model described in Section 2. We introduce a new parameter p_m ($m \hat{=} \text{monotone}$), which denotes the probability of a monotone combination of two partial hypotheses. Here, we formulate the recursion equation for a bigram language model, but of course, the same method can also be applied for a trigram lan-

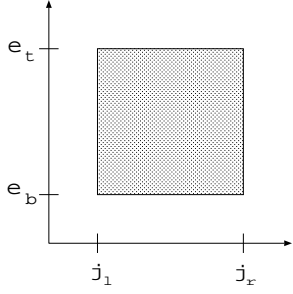


Figure 3: Illustration of the Q -table.

guage model.

$$Q_{j_l, j_r, e_b, e_t} = \max_{\substack{j_l \leq k < j_r, \\ e', e''}} \left\{ Q_{j_l, j_r, e_b, e_t}^0, \right. \\ \left. Q_{j_l, k, e_b, e'} \cdot Q_{k+1, j_r, e'', e_t} \cdot p(e''|e') \cdot p_m, \right. \\ \left. Q_{k+1, j_r, e_b, e'} \cdot Q_{j_l, k, e'', e_t} \cdot p(e''|e') \cdot (1 - p_m) \right\}$$

The resulting algorithm is similar to the CYK-parsing algorithm. It has a worst-case complexity of $\mathcal{O}(J^3 \cdot E^4)$. Here, J is the length of the source sentence and E is the vocabulary size of the target language.

3.2.2 Beam Search Algorithm

For the ITG constraints a dynamic programming search algorithm exists as described in the previous section. It would be more practical with respect to language model recombination to have an algorithm that generates the target sentence word by word or phrase by phrase. The idea is to start with the beam search decoder for unconstrained search and modify it in such a way that it will produce only reorderings that do not violate the ITG constraints. Now, we describe one way to obtain such a decoder. It has been pointed out in (Zens and Ney, 2003) that the ITG constraints can be characterized as follows: a reordering violates the ITG constraints if and only if it contains (3, 1, 4, 2) or (2, 4, 1, 3) as a subsequence. This means, if we select four columns and the corresponding rows from the alignment matrix and we obtain one of the two patterns illustrated in Figure 4, this reordering cannot be generated with the ITG constraints.

Now, we have to modify the beam search decoder such that it cannot produce these two patterns. We implement this in the following way. During the search, we have a coverage vector cov of the source sentence available for each partial hypothesis. A coverage vec-

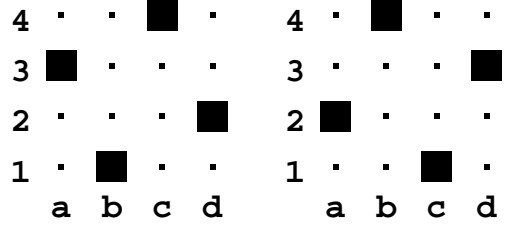


Figure 4: Illustration of the two reordering patterns that violate the ITG constraints.

tor is a binary vector marking the source sentence words that have already been translated (covered). Additionally, we know the current source sentence position j_c and a candidate source sentence position j_n to be translated next.

To avoid the patterns in Figure 4, we have to constrain the placement of the third phrase, because once we have placed the first three phrases we also have determined the position of the fourth phrase as the remaining uncovered position. Thus, we check the following constraints:

$$\text{case a) } j_n < j_c \quad (1)$$

$$\forall j_n < j < j_c : cov[j] \rightarrow cov[j + 1]$$

$$\text{case b) } j_c < j_n \quad (2)$$

$$\forall j_c < j < j_n : cov[j] \rightarrow cov[j - 1]$$

The constraints in Equations 1 and 2 enforce the following: imagine, we traverse the coverage vector cov from the current position j_c to the position to be translated next j_n . Then, it is not allowed to move from an uncovered position to a covered one.

Now, we sketch the proof that these constraints are equivalent to the ITG constraints. It is easy to see that the constraint in Equation 1 avoids the pattern on the left-hand side in Figure 4. To be precise: after placing the first two phrases at (b,1) and (d,2), it avoids the placement of the third phrase at (a,3). Similarly, the constraint in Equation 2 avoid the pattern on the right-hand side in Figure 4. Therefore, if we enforce the constraints in Equation 1 and Equation 2, we cannot violate the ITG constraints.

We still have to show that we can generate all the reorderings that do not violate the ITG constraints. Equivalently, we show that any reordering that violates the constraints in Equation 1 or Equation 2 will also violate the ITG constraints. It is rather easy to see that any reordering that violates the constraint in

Table 1: Statistics of the BTEC corpus.

		Japanese	English
train	Sentences	152 K	
	Words	1 044 K	893 K
	Vocabulary	17 047	12 020
dev	sentences	500	
	words	3 361	2 858
test	sentences	510	
	words	3 498	–

Table 2: Statistics of the SLDB corpus.

		Japanese	English
train	Sentences	15 K	
	Words	201 K	190 K
	Vocabulary	4 757	3 663
test	sentences	330	
	words	3 940	–

Equation 1 will generate the pattern on the left-hand side in Figure 4. The conditions to violate Equation 1 are the following: the new candidate position j_n is to the left of the current position j_c , e.g. positions (a) and (d). Somewhere in between there has to be an covered position j whose successor position $j + 1$ is uncovered, e.g. (b) and (c). Therefore, any reordering that violates Equation 1 generates the pattern on the left-hand side in Figure 4, thus it violates the ITG constraints.

4 Results

4.1 Corpus Statistics

To investigate the effect of reordering constraints, we have chosen two Japanese–English tasks, because the word order in Japanese and English is rather different. The first task is the *Basic Travel Expression Corpus* (BTEC) task (Takezawa et al., 2002). The corpus statistics are shown in Table 1. This corpus consists of phrasebook entries.

The second task is the *Spoken Language DataBase* (SLDB) task (Morimoto et al., 1994). This task consists of transcription of spoken dialogs in the domain of hotel reservation. Here, we use domain-specific training data in addition to the BTEC corpus. The corpus statistics of this additional corpus are shown in Table 2. The development corpus is the same for both tasks.

4.2 Evaluation Criteria

WER (word error rate). The WER is computed as the minimum number of substitution, insertion and deletion operations that have to

be performed to convert the generated sentence into the reference sentence.

PER (position-independent word error rate). A shortcoming of the WER is that it requires a perfect word order. The word order of an acceptable sentence can be different from that of the target sentence, so that the WER measure alone could be misleading. The PER compares the words in the two sentences ignoring the word order.

BLEU. This score measures the precision of unigrams, bigrams, trigrams and fourgrams with respect to a reference translation with a penalty for too short sentences (Papineni et al., 2002). The BLEU score measures accuracy, i.e. large BLEU scores are better.

NIST. This score is similar to BLEU. It is a weighted n -gram precision in combination with a penalty for too short sentences (Dodington, 2002). The NIST score measures accuracy, i.e. large NIST scores are better.

Note that for each source sentence, we have as many as 16 references available. We compute all the preceding criteria with respect to multiple references.

4.3 System Comparison

In Table 3 and Table 4, we show the translation results for the BTEC task. First, we observe that the overall quality is rather high on this task. The average length of the used alignment templates is about five source words in all systems. The monotone search (**mon**) shows already good performance on short sentences with less than 10 words. We conclude that for short sentences the reordering is captured within the alignment templates. On the other hand, the monotone search degrades for long sentences with at least 10 words resulting in a WER of 16.6% for these sentences.

We present the results for various nonmonotone search variants: the first one is with the IBM constraints (**skip**) as described in Section 3.1. We allow for skipping one or two phrases. Our experiments showed that if we set the maximum number of phrases to be skipped to three or more the translation results are equivalent to the search without any reordering constraints (**free**). The results for the ITG constraints as described in Section 3.2 are also presented.

The unconstrained reorderings improve the total translation quality down to a WER of 11.5%. We see that especially the long sentences benefit from the reorderings resulting in an improvement from 16.6% to 13.8%. Comparing the results for the free reorderings and

Table 3: Translation performance WER[%] for the BTEC task (510 sentences). Sentence lengths: short: < 10 words, long: \geq 10 words; times in milliseconds per sentence.

	WER[%]			time[ms]	
	sentence length				
reorder	short	long	all		
mon	11.4	16.6	12.7	73	
skip	1	10.8	13.5	11.4	134
	2	10.8	13.4	11.4	169
free	10.8	13.8	11.5	194	
ITG	10.6	12.2	11.0	164	

Table 4: Translation performance for the BTEC task (510 sentences).

	error rates[%]		accuracy measures		
	WER	PER	BLEU[%]	NIST	
reorder					
mon	12.7	10.6	86.8	14.14	
skip	1	11.4	10.1	88.0	14.19
	2	11.4	10.1	88.1	14.20
free	11.5	10.0	88.0	14.19	
ITG	11.0	9.9	88.2	14.25	

the ITG reorderings, we see that the ITG system always outperforms the unconstrained system. The improvement on the whole test set is statistically significant at the 95% level.¹

In Table 5 and Table 6, we show the results for the SLDB task. First, we observe that the overall quality is lower than for the BTEC task. The SLDB task is a spoken language translation task and the training corpus for spoken language is rather small. This is also reflected in the average length of the used alignment templates that is about three source words compared to about five words for the BTEC task.

The results on this task are similar to the results on the BTEC task. Again, the ITG constraints perform best. Here, the improvement compared to the unconstrained search is statistically significant at the 99% level. Compared to the monotone search, the BLEU score for the ITG constraints improves from 54.4% to 57.1%.

5 Related Work

Recently, phrase-based translation approaches became more and more popular. Marcu and Wong (2002) present a joint probability model for phrase-based translation. In (Koehn et

¹The statistical significance test were done for the WER using bootstrap resampling.

Table 5: Translation performance WER[%] for the SLDB task (330 sentences). Sentence lengths: short: < 10 words, long: \geq 10 words; times in milliseconds per sentence.

	WER[%]			time[ms]	
	sentence length				
reorder	short	long	all		
mon	32.0	52.6	48.1	911	
skip	1	31.9	51.1	46.9	3 175
	2	32.0	51.4	47.2	4 549
free	32.0	51.4	47.2	4 993	
ITG	31.8	50.9	46.7	4 472	

Table 6: Translation performance for the SLDB task (330 sentences).

	error rates[%]		accuracy measures		
	WER	PER	BLEU[%]	NIST	
reorder					
mon	48.1	35.5	54.4	9.45	
skip	1	46.9	35.0	56.8	9.71
	2	47.2	35.1	57.1	9.74
free	47.2	34.9	57.1	9.75	
ITG	46.7	34.6	57.1	9.76	

al., 2003), various aspects of phrase-based systems are compared, e.g. the phrase extraction method, the underlying word alignment model, or the maximum phrase length. In (Vogel, 2003), a phrase-based system is used that allows reordering within a window of up to three words. Improvements for a Chinese–English task are reported compared to a monotone search.

The ITG constraints were introduced in (Wu, 1995). The applications were, for instance, the segmentation of Chinese character sequences into Chinese words and the bracketing of the source sentence into sub-sentential chunks. Investigations on the IBM constraints (Berger et al., 1996) for single-word based statistical machine translation can be found e.g. in (Tillmann and Ney, 2003). A comparison of the ITG constraints and the IBM constraints for single-word based models can be found in (Zens and Ney, 2003). In this work, we investigated these reordering constraints for phrase-based statistical machine translation.

6 Conclusions

We have presented different reordering constraints for phrase-based statistical machine translation, namely the IBM constraints and the ITG constraints, as well as efficient dynamic programming algorithms. Translation results were reported for two Japanese–

English translation tasks. Both type of reordering constraints resulted in improvements compared to a monotone search. Restricting the reorderings according to the IBM constraints resulted already in a translation quality similar to an unconstrained search. The translation results with the ITG constraints even outperformed the unconstrained search consistently on all error criteria. The improvements have been found statistically significant.

The ITG constraints showed the best performance on both tasks. Therefore we plan to further improve this method. Currently, the probability model for the ITG constraints is very simple. More sophisticated models, such as phrase dependent inversion probabilities, might be promising.

Acknowledgments

This work was partially done at the Spoken Language Translation Research Laboratories (SLT) at the Advanced Telecommunication Research Institute International (ATR), Kyoto, Japan. This research was supported in part by the Telecommunications Advancement Organization of Japan. This work has been partially funded by the EU project PF-Star, IST-2001-37599.

References

- A. L. Berger, P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, J. R. Gillett, A. S. Kehler, and R. L. Mercer. 1996. Language translation apparatus and method of using context-based translation models, United States patent, patent number 5510981, April.
- P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June.
- G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. ARPA Workshop on Human Language Technology*.
- K. Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615, December.
- P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pages 127–133, Edmonton, Canada, May/June.
- D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. Conf. on Empirical Methods for Natural Language Processing*, pages 133–139, Philadelphia, PA, July.
- T. Morimoto, N. Uratani, T. Takezawa, O. Furuse, Y. Sobashima, H. Iida, A. Nakamura, Y. Sagisaka, N. Higuchi, and Y. Yamazaki. 1994. A speech and language database for speech translation research. In *Proc. of the 3rd Int. Conf. on Spoken Language Processing (ICSLP'94)*, pages 1791–1794, Yokohama, Japan, September.
- F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July.
- F. J. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, College Park, MD, June.
- F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.
- K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA, July.
- T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proc. of the Third Int. Conf. on Language Resources and Evaluation (LREC)*, pages 147–152, Las Palmas, Spain, May.
- C. Tillmann and H. Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133, March.
- S. Vogel. 2003. SMT decoder dissected: Word reordering. In *Proc. of the Int. Conf. on Natural Language Processing and Knowledge Engineering (NLP-KE)*, pages 561–566, Beijing, China, October.
- D. Wu. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proc. of the 14th International Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1328–1334, Montreal, August.
- D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, September.
- R. Zens and H. Ney. 2003. A comparative study on reordering constraints in statistical machine translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 144–151, Sapporo, Japan, July.