# A Comparative Study on Reordering Constraints in Statistical Machine Translation

**Richard Zens** and **Hermann Ney**
Chair of Computer Science VI
RWTH Aachen - University of Technology
{zens,ney}@cs.rwth-aachen.de

## Abstract

In statistical machine translation, the generation of a translation hypothesis is computationally expensive. If arbitrary word-reorderings are permitted, the search problem is NP-hard. On the other hand, if we restrict the possible word-reorderings in an appropriate way, we obtain a polynomial-time search algorithm.

In this paper, we compare two different reordering constraints, namely the ITG constraints and the IBM constraints. This comparison includes a theoretical discussion on the permitted number of reorderings for each of these constraints. We show a connection between the ITG constraints and the since 1870 known *Schröder* numbers.

We evaluate these constraints on two tasks: the Verbmobil task and the Canadian Hansards task. The evaluation consists of two parts: First, we check how many of the Viterbi alignments of the training corpus satisfy each of these constraints. Second, we restrict the search to each of these constraints and compare the resulting translation hypotheses.

The experiments will show that the baseline ITG constraints are not sufficient on the Canadian Hansards task. Therefore, we present an extension to the ITG constraints. These extended ITG constraints increase the alignment coverage from about 87% to 96%.

## 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:

$$\hat{e}_1^I = \operatorname*{argmax}_{e_1^I} \{Pr(e_1^I|f_1^J)\} \quad (1)$$

$$= \operatorname*{argmax}_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)\} \quad (2)$$

The decomposition into two knowledge sources in Eq. 2 is the so-called 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.

In this paper, we will focus on the alignment problem, i.e. the mapping between source sentence positions and target sentence positions. As the word order in source and target language may differ, the search algorithm has to allow certain word-reorderings. If arbitrary word-reorderings are allowed, the search problem is NP-hard (Knight, 1999). Therefore, we have to restrict the possible reorderings in some way to make the search problem feasible. Here, we will discuss two such constraints in detail. The first constraints are based on *inversion transduction grammars* (ITG) (Wu, 1995; Wu, 1997). In the following, we will call these the ITG constraints. The second constraints are the IBM constraints (Berger et al., 1996). In the next section, we will describe these constraints from a theoretical point of view. Then, we will describe the resulting search algorithm and its extension for word graph generation. Afterwards, we will analyze the Viterbi alignments produced during the training of the alignment models. Then, we will compare the translation results when restricting the search to either of these constraints.

## 2 Theoretical Discussion

In this section, we will discuss the reordering constraints from a theoretical point of view. We will answer the question of how many word-reorderings are permitted for the ITG constraints as well as for the IBM constraints. Since we are only interested in the number of possible reorderings, the specific word identities are of no importance here. Furthermore, we assume a one-to-one correspondence between source and target words. Thus, we are interested in the number of word-reorderings, i.e. permutations, that satisfy the chosen constraints. First, we will consider the ITG constraints. Afterwards, we will describe the IBM constraints.

### 2.1 ITG Constraints

Let us now consider the ITG constraints. Here, we interpret the input sentence as a sequence of blocks. In the beginning, each position is a block of its own. Then, the permutation process can be seen as follows: we select two consecutive blocks and merge them to a single block by choosing between two options: either keep them in monotone order or invert the order. This idea is illustrated in Fig. 1. The white boxes represent the two blocks to be merged.

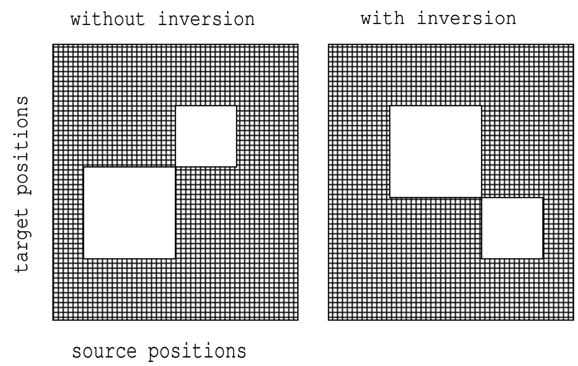


Figure 1: Illustration of monotone and inverted concatenation of two consecutive blocks.

Now, we investigate, how many permutations are obtainable with this method. A permutation derived by the above method can be represented as a binary tree where the inner nodes are colored either black or white. At black nodes the resulting sequences of the children are inverted. At white nodes they are kept in monotone order. This representation is equivalent to the parse trees of the simple grammar in (Wu, 1997).

We observe that a given permutation may be constructed in several ways by the above method. For instance, let us consider the identity permutation of $1, 2, ..., n$. Any binary tree with $n$ nodes and all inner nodes colored white (monotone order) is a possible representation of this permutation. To obtain a unique representation, we pose an additional constraint on the binary trees: if the right son of a node is an inner node, it has to be colored with the opposite color. With this constraint, each of these binary trees is unique and equivalent to a parse tree of the 'canonical-form' grammar in (Wu, 1997).

In (Shapiro and Stephens, 1991), it is shown that the number of such binary trees with $n$ nodes is the $(n-1)th$ large *Schröder* number $S_{n-1}$. The (small) *Schröder* numbers have been first described in (Schröder, 1870) as the number of bracketings of a given sequence (Schröder's second problem). The large *Schröder* numbers are just twice the *Schröder* numbers. Schröder remarked that the ratio between two consecutive *Schröder* numbers approaches $3 + 2\sqrt{2} = 5.8284...$ . A second-order recurrence for the large *Schröder* numbers is:

$$(n+1)S_n \quad = \quad 3(2n-1)S_{n-1} - (n-2)S_{n-2}$$

with $n \geq 2$ and $S_0 = 1$, $S_1 = 2$.

The *Schröder* numbers have many combinatorical interpretations. Here, we will mention only two of them. The first one is another way of viewing at the ITG constraints. The number of permutations of the sequence $1, 2, ..., n$, which avoid the subsequences $(3, 1, 4, 2)$ and $(2, 4, 1, 3)$, is the large *Schröder* number $S_{n-1}$. More details on forbidden subsequences can be found in (West, 1995). The interesting point is that a search with the ITG constraints cannot generate a word-reordering that contains one of these two subsequences. In (Wu, 1997), these forbidden subsequences are called 'inside-out' transpositions.

Another interpretation of the *Schröder* numbers is given in (Knuth, 1973): The number of permutations that can be sorted with an output-restricted double-ended queue (deque) is exactly the large *Schröder* number. Additionally, Knuth presents an approximation for the large *Schröder* numbers:

$$S_n \approx c \cdot (3+\sqrt{8})^n \cdot n^{-\frac{3}{2}} \tag{3}$$

where $c$ is set to $\frac{1}{2}\sqrt{(3\sqrt{2}-4)/\pi}$. This approximation function confirms the result of Schröder, and we obtain $S_n \in \Theta((3+\sqrt{8})^n)$, i.e. the *Schröder* numbers grow like $(3+\sqrt{8})^n \approx 5.83^n$.

### 2.2 IBM Constraints

In this section, we will describe the IBM constraints (Berger et al., 1996). Here, we mark each position in the source sentence either as covered or uncovered. In the beginning, all source positions are uncovered. Now, the target sentence is produced from bottom to top. A target position must be aligned to one of the first $k$ uncovered source positions. The IBM constraints are illustrated in Fig. 2.

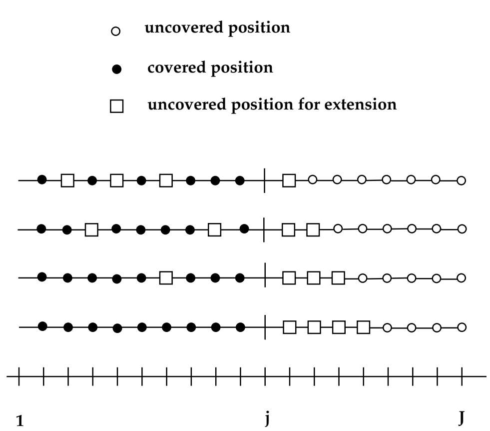


Figure 2: Illustration of the IBM constraints.

For most of the target positions there are $k$ permitted source positions. Only towards the end of the sentence this is reduced to the number of remaining uncovered source positions. Let $n$ denote the length of the input sequence and let $r_n$ denote the permitted number of permutations with the IBM constraints. Then, we obtain:

$$r_n = \begin{cases} k^{n-k} \cdot k! & n > k \\ n! & n \leq k \end{cases} \tag{4}$$

Typically, $k$ is set to 4. In this case, we obtain an asymptotic upper and lower bound of $4^n$, i.e. $r_n \in \Theta(4^n)$.

In Tab. 1, the ratio of the number of permitted reorderings for the discussed constraints is listed as a function of the sentence length. We see that for longer sentences the ITG constraints allow for more reorderings than the IBM constraints. For sentences of length 10 words, there are about twice as many reorderings for the ITG constraints than for the IBM constraints. This ratio steadily increases. For longer sentences, the ITG constraints allow for much more flexibility than the IBM constraints.

## 3 Search

Now, let us get back to more practical aspects. Reordering constraints are more or less useless, if they do not allow the maximization of Eq. 2 to be performed in an efficient way. Therefore, in this section, we will describe different aspects of the search algorithm for the ITG constraints. First, we will present the dynamic programming equations and the resulting complexity. Then, we will describe pruning techniques to accelerate the search. Finally, we will extend the basic algorithm for the generation of word graphs.

### 3.1 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 Fig. 3.

Here, we initialize this table with monotone translations of IBM Model 4. Therefore, $Q^0_{j_l,j_r,e_b,e_t}$ denotes the probability of the best monotone hypothesis of IBM Model 4. Alternatively, we could use any other single-word based lexicon as well as phrase-based models for this initialization. Our choice is the IBM Model4 to make the results as comparable

Table 1: Ratio of the number of permitted reorderings with the ITG constraints $S_{n-1}$ and the IBM constraints $r_n$ for different sentence lengths $n$.

| $n$ | 1 ... 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_{n-1}/r_n$ | $\approx$ 1.0 | 1.2 | 1.4 | 1.7 | 2.1 | 2.6 | 3.4 | 4.3 | 5.6 | 7.4 | 9.8 | 13.0 | 17.4 | 23.3 | 31.4 |

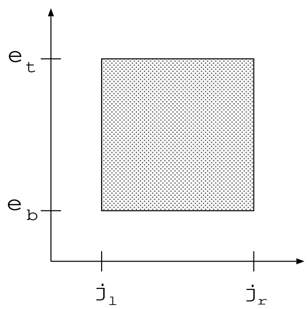


Figure 3: Illustration of the $Q$-table.

as possible to the search with the IBM constraints. We introduce a new parameter $p_m$ ($m \hat{=}$ monotone), which denotes the probability of a monotone combination of two partial hypotheses.

$$
\begin{aligned}
Q_{j_l,j_r,e_b,e_t} = & \qquad\qquad\qquad\qquad\qquad\qquad\qquad (5) \\
\max_{\substack{j_l \le k < j_r, \\ e', e''}} & \Big\{ Q^0_{j_l,j_r,e_b,e_t}, \\
& Q_{j_l,k,e_b,e'} \cdot Q_{k+1,j_r,e'',e_t} \cdot p(e''|e') \cdot p_m, \\
& Q_{k+1,j_r,e_b,e'} \cdot Q_{j_l,k,e'',e_t} \cdot p(e''|e') \cdot (1 - p_m) \Big\}
\end{aligned}
$$

We formulated this equation for a bigram language model, but of course, the same method can also be applied for a trigram language model. 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 Pruning

Although the described search algorithm has a polynomial-time complexity, even with a bigram language model the search space is very large. A full search is possible but time consuming. The situation gets even worse when a trigram language model is used. Therefore, pruning techniques are obligatory to reduce the translation time.

Pruning is applied to hypotheses that translate the same subsequence $f_{j_l}^{j_r}$ of the source sentence. We use pruning in the following two ways. The first pruning technique is histogram pruning: we restrict the number of translation hypotheses per sequence $f_{j_l}^{j_r}$. For each sequence $f_{j_l}^{j_r}$, we keep only a fixed number of translation hypotheses. The second pruning technique is threshold pruning: the idea is to remove all hypotheses that have a low probability relative to the best hypothesis. Therefore, we introduce a threshold pruning parameter $q$, with $0 \le q \le 1$. Let $Q^*_{j_l,j_r}$ denote the maximum probability of all translation hypotheses for $f_{j_l}^{j_r}$. Then, we prune a hypothesis iff:

$$Q_{j_l,j_r,e_b,e_t} < q \cdot Q^*_{j_l,j_r}$$

Applying these pruning techniques the computational costs can be reduced significantly with almost no loss in translation quality.

### 3.3 Generation of Word Graphs

The generation of word graphs for a bottom-top search with the IBM constraints is described in (Ueffing et al., 2002). These methods cannot be applied to the CYK-style search for the ITG constraints. Here, the idea for the generation of word graphs is the following: assuming we already have word graphs for the source sequences $f_{j_l}^{k}$ and $f_{k+1}^{j_r}$, then we can construct a word graph for the sequence $f_{j_l}^{j_r}$ by concatenating the partial word graphs either in monotone or inverted order.

Now, we describe this idea in a more formal way. A word graph is a directed acyclic graph (dag) with one start and one end node. The edges are annotated with target language words or phrases. We also allow $\epsilon$-transitions. These are edges annotated with the empty word. Additionally, edges may be annotated with probabilities of the language or translation model. Each path from start node to end node represents one translation hypothesis. The probability of this hypothesis is calculated by multiplying the probabilities along the path.

During the search, we have to combine two word graphs in either monotone or inverted order. This is done in the following way: we are given two word graphs $w_1$ and $w_2$ with start and end nodes $(s_1, g_1)$ and $(s_2, g_2)$, respectively. First, we add an $\epsilon$-transition $(g_1, s_2)$ from the end node of the first graph $w_1$ to the start node of the second graph $w_2$ and annotate this edge with the probability of a monotone concatenation $p_m$. Second, we create a copy of each of the original word graphs $w_1$ and $w_2$. Then, we add an $\epsilon$-transition $(g_2, s_1)$ from the end node of the copied second graph to the start node of the copied first graph. This edge is annotated with the probability of a inverted concatenation $1 - p_m$. Now, we have obtained two word graphs: one for a monotone and one for a inverted concatenation. The final word graphs is constructed by merging the two start nodes and the two end nodes, respectively.

Let $W(j_l, j_r)$ denote the word graph for the source sequence $f_{j_l}^{j_r}$. This graph is constructed from the word graphs of all subsequences of $(j_l, j_r)$. Therefore, we assume, these word graphs have already been produced. For all source positions $k$ with $j_l \leq k < j_r$, we combine the word graphs $W(j_l, k)$ and $W(k+1, j_r)$ as described above. Finally, we merge all start nodes of these graphs as well as all end nodes. Now, we have obtained the word graph $W(j_l, j_r)$ for the source sequence $f_{j_l}^{j_r}$. As initialization, we use the word graphs of the monotone IBM4 search.

### 3.4 Extended ITG constraints

In this section, we will extend the ITG constraints described in Sec. 2.1. This extension will go beyond basic reordering constraints.

We already mentioned that the use of consecutive phrases within the ITG approach is straightforward. The only thing we have to change is the initialization of the $Q$-table. Now, we will extend this idea to phrases that are non-consecutive in the source language. For this purpose, we adopt the view of the ITG constraints as a bilingual grammar as, e.g., in (Wu, 1997). For the baseline ITG constraints, the resulting grammar is:

$$A \rightarrow [AA] \mid \langle AA \rangle \mid f/e \mid f/\epsilon \mid \epsilon/e$$

Here, $[AA]$ denotes a monotone concatenation and $\langle AA \rangle$ denotes an inverted concatenation.

Let us now consider the case of a source phrase consisting of two parts $f_1$ and $f_2$. Let $e$ denote the corresponding target phrase. We add the productions

$$A \rightarrow [e/f_1 \; A \; \epsilon/f_2] \mid \langle e/f_1 \; A \; \epsilon/f_2 \rangle$$

to the grammar. The probabilities of these productions are, dependent on the translation direction, $p(e|f_1, f_2)$ or $p(f_1, f_2|e)$, respectively. Obviously, these productions are not in the normal form of an ITG, but with the method described in (Wu, 1997), they can be normalized.

## 4 Corpus Statistics

In the following sections we will present results on two tasks. Therefore, in this section we will show the corpus statistics for each of these tasks.

### 4.1 Verbmobil

The first task we will present results on is the Verbmobil task (Wahlster, 2000). The domain of this corpus is appointment scheduling, travel planning, and hotel reservation. It consists of transcriptions of spontaneous speech. Table 2 shows the corpus statistics of this corpus. The training corpus (Train) was used to train the IBM model parameters. The remaining free parameters, i.e. $p_m$ and the model scaling factors (Och and Ney, 2002), were adjusted on the development corpus (Dev). The resulting system was evaluated on the test corpus (Test).

Table 2: Statistics of training and test corpus for the Verbmobil task (PP=perplexity, SL=sentence length).

| | | German | English |
|---|---|---|---|
| Train | Sentences | 58 073 | |
| | Words | 519 523 | 549 921 |
| | Vocabulary | 7 939 | 4 672 |
| | Singletons | 3 453 | 1 698 |
| | average SL | 8.9 | 9.5 |
| Dev | Sentences | 276 | |
| | Words | 3 159 | 3 438 |
| | Trigram PP | - | 28.1 |
| | average SL | 11.5 | 12.5 |
| Test | Sentences | 251 | |
| | Words | 2 628 | 2 871 |
| | Trigram PP | - | 30.5 |
| | average SL | 10.5 | 11.4 |

Table 3: Statistics of training and test corpus for the Canadian Hansards task (PP=perplexity, SL=sentence length).

| | | French | English |
|---|---|---|---|
| Train | Sentences | 1.5M | |
| | Words | 24M | 22M |
| | Vocabulary | 100 269 | 78 332 |
| | Singletons | 40 199 | 31 319 |
| | average SL | 16.6 | 15.1 |
| Test | Sentences | 5432 | |
| | Words | 97 646 | 88 773 |
| | Trigram PP | – | 179.8 |
| | average SL | 18.0 | 16.3 |

### 4.2 Canadian Hansards

Additionally, we carried out experiments on the Canadian Hansards task. This task contains the proceedings of the Canadian parliament, which are kept by law in both French and English. About 3 million parallel sentences of this bilingual data have been made available by the Linguistic Data Consortium (LDC). Here, we use a subset of the data containing only sentences with a maximum length of 30 words. Table 3 shows the training and test corpus statistics.

## 5 Evaluation in Training

In this section, we will investigate for each of the constraints the coverage of the training corpus alignment. For this purpose, we compute the Viterbi alignment of IBM Model 5 with GIZA++ (Och and Ney, 2000). This alignment is produced without any restrictions on word-reorderings. Then, we check for every sentence if the alignment satisfies each of the constraints. The ratio of the number of satisfied alignments and the total number of sentences is referred to as coverage. Tab. 4 shows the results for the Verbmobil task and for the Canadian Hansards task. It contains the results for both translation directions German-English (S→T) and English-German (T→S) for the Verbmobil task and French-English (S→T) and English-French (T→S) for the Canadian Hansards task, respectively.

For the Verbmobil task, the baseline ITG constraints and the IBM constraints result in a similar coverage. It is about 91% for the German-English translation direction and about 88% for the English-German translation direction. A significantly higher coverage of about 96% is obtained with the extended ITG constraints. Thus with the extended ITG constraints, the coverage increases by about 8% absolute.

Table 4: Coverage on the training corpus for alignment constraints for the Verbmobil task (VM) and for the Canadian Hansards task (CH).

| | | | coverage [%] | |
|---|---|---|---|---|
| task | constraint | | S→T | T→S |
| VM | IBM | | 91.0 | 88.1 |
| | ITG | baseline | 91.6 | 87.0 |
| | | extended | 96.5 | 96.9 |
| CH | IBM | | 87.1 | 86.7 |
| | ITG | baseline | 81.3 | 73.6 |
| | | extended | 96.1 | 95.6 |

For the Canadian Hansards task, the baseline ITG constraints yield a worse coverage than the IBM constraints. Especially for the English-French translation direction, the ITG coverage of 73.6% is very low. Again, the extended ITG constraints obtained the best results. Here, the coverage increases from about 87% for the IBM constraints to about 96% for the extended ITG constraints.

## 6 Translation Experiments

### 6.1 Evaluation Criteria

In our experiments, we use the following error 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 target sentence.

- PER (position-independent word error rate):
  A shortcoming of the WER is the fact that it requires a perfect word order. The PER compares the words in the two sentences ignoring the word order.

- mWER (multi-reference word error rate):
  For each test sentence, not only a single reference translation is used, as for the WER, but a whole set of reference translations. For each translation hypothesis, the WER to the most similar sentence is calculated (Nießen et al., 2000).

- BLEU score:
  This score measures the precision of unigrams, bigrams, trigrams and fourgrams with respect to a whole set of reference translations with a penalty for too short sentences (Papineni et al., 2001). BLEU measures accuracy, i.e. large BLEU scores are better.

- SSER (subjective sentence error rate):
  For a more detailed analysis, subjective judgments by test persons are necessary. Each translated sentence was judged by a human examiner according to an error scale from 0.0 to 1.0 (Nießen et al., 2000).

### 6.2 Translation Results

In this section, we will present the translation results for both the IBM constraints and the baseline ITG constraints. We used a single-word based search with IBM Model 4. The initialization for the ITG constraints was done with monotone IBM Model 4 translations. So, the only difference between the two systems are the reordering constraints.

In Tab. 5 the results for the Verbmobil task are shown. We see that the results on this task are similar. The search with the ITG constraints yields slightly lower error rates.

Some translation examples of the Verbmobil task are shown in Tab. 6. We have to keep in mind, that the Verbmobil task consists of transcriptions of *spontaneous* speech. Therefore, the source sentences as well as the reference translations may have an unorthodox grammatical structure. In the first example, the German verb-group ("würde vorschlagen") is split into two parts. The search with the ITG constraints is able to produce a correct translation. With the IBM constraints, it is not possible to translate this verb-group correctly, because the distance between the two parts is too large (more than four words). As we see in the second example, in German the verb of a subordinate clause is placed at the end ("übernachten"). The IBM search is not able to perform the necessary long-range reordering, as it is done with the ITG search.

## 7 Related Work

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. In (Wu, 1996) the baseline ITG constraints were used for statistical machine translation. The resulting algorithm is similar to the one presented in Sect. 3.1, but here, we use monotone translation hypotheses of the full IBM Model 4 as initialization, whereas in (Wu, 1996) a single-word based lexicon model is used. In (Vilar, 1998) a model similar to Wu's method was considered.

## 8 Conclusions

We have described the ITG constraints in detail and compared them to the IBM constraints. We draw the following conclusions: especially for long sentences the ITG constraints allow for higher flexibility in word-reordering than the IBM constraints. Regarding the Viterbi alignment in training, the baseline ITG constraints yield a similar coverage as the IBM constraints on the Verbmobil task. On the Canadian Hansards task the baseline ITG constraints were not sufficient. With the extended ITG constraints the coverage improves significantly on both tasks. On the Canadian Hansards task the coverage increases from about 87% to about 96%.

We have presented a polynomial-time search algorithm for statistical machine translation based on the ITG constraints and its extension for the generation of word graphs. We have shown the translation results for the Verbmobil task. On this task, the translation quality of the search with the baseline ITG constraints is already competitive with the results for the IBM constraints. Therefore, we expect the search with the extended ITG constraints to outperform the search with the IBM constraints.

Future work will include the automatic extraction of the bilingual grammar as well as the use of this grammar for the translation process.

## 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

Table 5: Translation results on the Verbmobil task.

| type | automatic | | | | human |
|---|---|---|---|---|---|
| System | WER [%] | PER [%] | mWER [%] | BLEU [%] | SSER [%] |
| IBM | 46.2 | 33.3 | 40.0 | 42.5 | 40.8 |
| ITG | 45.6 | 33.9 | 40.0 | 37.1 | 42.0 |

Table 6: Verbmobil: translation examples.

| | |
|---|---|
| source | ja, ich würde den Flug um viertel nach sieben vorschlagen. |
| reference | yes, I would suggest the flight at a quarter past seven. |
| ITG | yes, I would suggest the flight at seven fifteen. |
| IBM | yes, I would be the flight at quarter to seven suggestion. |

| | |
|---|---|
| source | ich schlage vor, dass wir in Hannover im Hotel Grünschnabel übernachten. |
| reference | I suggest to stay at the hotel Grünschnabel in Hanover. |
| ITG | I suggest that we stay in Hanover at hotel Grünschnabel. |
| IBM | I suggest that we are in Hanover at hotel Grünschnabel stay. |

translation. *Computational Linguistics*, 16(2):79–85, June.

K. Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615, December.

D. E. Knuth. 1973. *The Art of Computer Programming*, volume 1 - Fundamental Algorithms. Addison-Wesley, Reading, MA, 2nd edition.

S. Nießen, F. J. Och, G. Leusch, and H. Ney. 2000. An evaluation tool for machine translation: Fast evaluation for MT research. In *Proc. of the Second Int. Conf. on Language Resources and Evaluation (LREC)*, pages 39–45, Athens, Greece, May.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–447, Hong Kong, October.

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, July.

K. A. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, September.

E. Schröder. 1870. Vier combinatorische Probleme. *Zeitschrift für Mathematik und Physik*, 15:361–376.

L. Shapiro and A. B. Stephens. 1991. Boostrap percolation, the Schröder numbers, and the $n$-kings problem. *SIAM Journal on Discrete Mathematics*, 4(2):275–280, May.

N. Ueffing, F. J. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. In *Proc. Conf. on Empirical Methods for Natural Language Processing*, pages 156–163, Philadelphia, PA, July.

J. M. Vilar. 1998. *Aprendizaje de Transductores Subsecuenciales para su empleo en tareas de Dominio Restringido*. Ph.D. thesis, Universidad Politecnica de Valencia.

W. Wahlster, editor. 2000. *Verbmobil: Foundations of speech-to-speech translations*. Springer Verlag, Berlin, Germany, July.

J. West. 1995. Generating trees and the Catalan and Schröder numbers. *Discrete Mathematics*, 146:247–262, November.

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. 1996. A polynomial-time algorithm for statistical machine translation. In *Proc. of the 34th Annual Conf. of the Association for Computational Linguistics (ACL '96)*, pages 152–158, Santa Cruz, CA, June.

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