Word Re-ordering and DP-based Search in Statistical Machine Translation

Christoph Tillmann and **Hermann Ney**
Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen - University of Technology
D-52056 Aachen, Germany
{tillmann,ney}@informatik.rwth-aachen.de

Abstract

In this paper, we describe a search procedure for statistical machine translation (MT) based on dynamic programming (DP). Starting from a DP-based solution to the traveling salesman problem, we present a novel technique to restrict the possible word reordering between source and target language in order to achieve an efficient search algorithm. A search restriction especially useful for the translation direction from German to English is presented. The experimental tests are carried out on the Verbmobil task (German-English, 8000-word vocabulary), which is a limited-domain spoken-language task.

1 Introduction

The goal of machine translation is the translation of a text given in some source language into a target language. We are given a source string $f_1^J = f_1 ... f_j ... f_J$ of length J, which is to be translated into a target string $e_1^I = e_1 ... e_i ... e_I$ of length I. Among all possible target strings, we will choose the string with the highest probability:

$$\begin{aligned} \hat{e}_1^I &= \arg\max_{e_1^I} \{Pr(e_1^I | f_1^J)\} \\ &= \arg\max_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)\} \quad . \quad (1) \end{aligned}$$

The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. $Pr(e_1^I)$ is the *language model* of the target language, whereas $Pr(f_1^J | e_1^I)$ is the *translation model*. Our approach uses word-to-word dependencies between source and target words. The model is often further restricted so that each source word is assigned to exactly one target word (Brown et al., 1993; Ney et al., 2000). These alignment models are similar to the concept of hidden Markov models (HMM) in speech recognition. The alignment mapping is $j \rightarrow i = a_j$ from source position j to target position $i = a_j$. The use of this alignment model raises major problems if a source word has to be aligned to several target words, e.g. when translating German compound nouns. A simple extension will be used to handle this problem.

In Section 2, we briefly review our approach to statistical machine translation. In Section 3, we introduce our novel concept to word re-ordering and a DP-based search, which is especially suitable for the translation direction from German to English. This approach is compared to another re-ordering scheme presented in (Berger et al., 1996). In Section 4, we present the performance measures used and give translation results on the Verbmobil task.

2 Basic Approach

In this section, we briefly review our translation approach. In Eq. (1), $Pr(e_1^I)$ is the language model, which is a trigram language model in this case. For the translation model $Pr(f_1^J | e_1^I)$, we go on the assumption that each source word is aligned to exactly one target word. The alignment model uses two kinds of parameters: alignment probabilities $p(a_j | a_{j-1}, I, J)$, where the probability of alignment a_j for position j depends on the previous alignment position a_{j-1} (Ney et al., 2000) and lexicon probabilities $p(f_j | e_{a_j})$. When aligning the words in parallel texts (for language pairs like Spanish-English, French-English, Italian-German,...), we typically observe a strong localization effect. In many cases, there is an even stronger restriction: over large portions of the source string, the alignment is monotone.

2.1 Inverted Alignments

To explicitly handle the word re-ordering between words in source and target language, we use the concept of the so-called inverted alignments as given in (Ney et al., 2000). An inverted alignment is defined as follows:

$$\text{inverted alignment:} \quad i \rightarrow j = b_i .$$

Target positions i are mapped to source positions b_i. What is important and is not expressed by the notation is the so-called coverage constraint: each source position j should be 'hit' exactly once by the path of the inverted alignment $b_1^I = b_1 ... b_i ... b_I$. Using the inverted alignments in the maximum approximation, we obtain as search criterion:

$$\max_I \left\{ p(J|I) \cdot \max_{e_1^I} \left\{ \prod_{i=1}^{I} p(e_i|e_{i-2}^{i-1}) \cdot \right.\right.$$

$$\left.\left. \cdot \max_{b_1^I} \prod_{i=1}^{I} [p(b_i|b_{i-1}, I, J) \cdot p(f_{b_i}|e_i)] \right\} \right\} =$$

$$= \max_I \left\{ p(J|I) \cdot \max_{e_1^I, b_1^I} \left\{ \prod_{i=1}^{I} [p(e_i|e_{i-2}^{i-1}) \cdot \right.\right.$$

$$\left.\left. \cdot p(b_i|b_{i-1}, I, J) \cdot p(f_{b_i}|e_i)] \right\} \right\},$$

where the two products over i have been merged into a single product over i. $p(e_i|e_{i-2}^{i-1})$ is the trigram language model probability. The inverted alignment probability $p(b_i|b_{i-1}, I, J)$ and the lexicon probability $p(f_{b_i}|e_i)$ are obtained by relative frequency estimates from the Viterbi alignment path after the final training iteration. The details are given in (Och and Ney, 2000). The sentence length probability $p(J|I)$ is omitted without any loss in performance. For the inverted alignment probability $p(b_i|b_{i-1}, I, J)$, we drop the dependence on the target sentence length I.

2.2 Word Joining

The baseline alignment model does not permit that a source word is aligned to two or more target words, e.g. for the translation direction from German to English, the German compound noun *'Zahnarzttermin'* causes problems, because it must be translated by the two target words *dentist's appointment*. We use a solution to this problem similar to the one presented in (Och et al., 1999), where target words are joined during training. The word joining is done on the basis of a likelihood criterion. An extended lexicon model is defined, and its likelihood is compared to a baseline lexicon model, which takes only single-word dependencies into account. E.g. when *'Zahnarzttermin'* is aligned to *dentist's*, the extended lexicon model might learn that *'Zahnarzttermin'* actually has to be aligned to both *dentist's* and *appointment*. In the following, we assume that this word joining has been carried out.

3 DP Algorithm for Statistical Machine Translation

In order to handle the necessary word re-ordering as an optimization problem within our dynamic programming approach, we describe a solution to the traveling salesman problem (TSP) which is based on dynamic programming (Held, Karp, 1962). The traveling salesman problem is an optimization problem which is defined as follows: given are a set of cities $S = s_1, \cdots, s_n$ and for each pair of cities s_i, s_j the cost $d_{ij} > 0$ for traveling from city s_i to city s_j. We are looking for the shortest tour visiting all cities exactly once while starting and ending in city s_1. A straightforward way to find the shortest tour is by trying all possible permutations of the n cities. The resulting algorithm has a complexity of $O(n!)$. However, dynamic programming can be used to find the shortest tour in exponential time, namely in $O(n^2 \cdot 2^n)$, using the algorithm by Held and Karp. The approach recursively evaluates a quantity $Q(\mathcal{C}, j)$, where $\mathcal{C}$ is the set of already visited cities and s_j is the last visited city. Subsets C of increasing cardinality c are processed. The algorithm works due to the fact that not all permutations of cities have to be considered explicitly. For a given partial hypothesis $(\mathcal{C}, j)$, the order in which the cities in $\mathcal{C}$ have been visited can be ignored (except j), only the score for the best path reaching j has to be stored. This algorithm can be applied to statistical machine translation. Using the concept of inverted alignments, we explicitly take care of the coverage constraint by introducing a coverage set $\mathcal{C}$ of source sentence positions that have been already processed. The advantage is that we can recombine search hypotheses by dynamic programming. The cities of the traveling salesman problem correspond to source

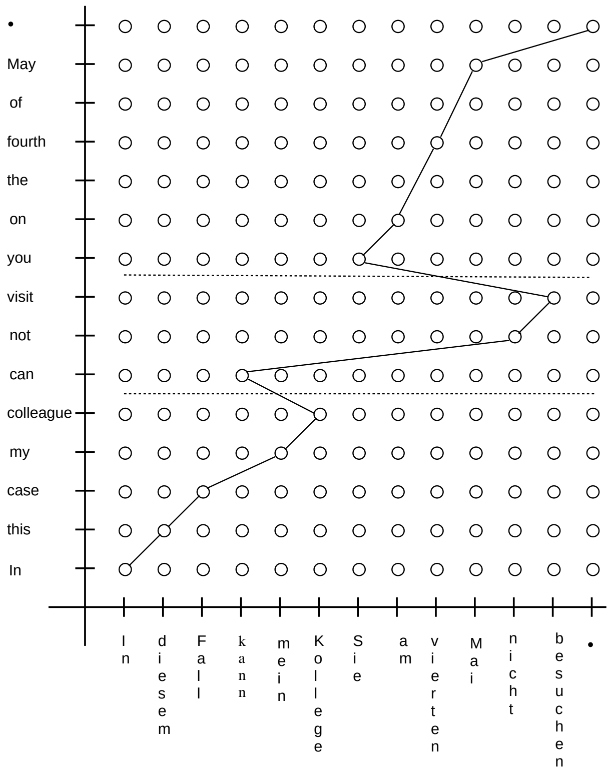

Figure 1: Re-ordering for the German verbgroup.

Table 1: DP algorithm for statistical machine translation.

input: source string $f_1 ... f_j ... f_J$

initialization

for each cardinality $c = 1, 2, \cdots, J$ do

  for each pair $(\mathcal{C}, j)$, where $j \in \mathcal{C}$ and $|\mathcal{C}| = c$ do

    for each target word $e \in E$

      $Q_{e'}(e, \mathcal{C}, j) = p(f_j|e) \max\limits_{\substack{\delta, e'' \\ j' \in \mathcal{C} \setminus \{j\}}} \{p(j|j', J) \cdot p(\delta) \cdot p_\delta(e|e', e'') \cdot Q_{e''}(e', \mathcal{C} \setminus \{j\}, j')\}$

words f_j in the input string of length J. For the final translation each source position is considered exactly once. Subsets of partial hypotheses with coverage sets $\mathcal{C}$ of increasing cardinality c are processed. For a trigram language model, the partial hypotheses are of the form $(e', e, \mathcal{C}, j)$. e', e are the last two target words, $\mathcal{C}$ is a coverage set for the already covered source positions and j is the last position visited. Each distance in the traveling salesman problem now corresponds to the negative logarithm of the product of the translation, alignment and language model probabilities. The following auxiliary quantity is defined:

$Q_{e'}(e, \mathcal{C}, j) :=$ probability of the best partial hypothesis (e_1^i, b_1^i), where $\mathcal{C} = \{b_k | k = 1, \cdots, i\}$, $b_i = j$, $e_i = e$ and $e_{i-1} = e'$.

The type of alignment we have considered so far requires the same length for source and target sentence, i.e. $I = J$. Evidently, this is an unrealistic assumption, therefore we extend the concept of inverted alignments as follows: When adding a new position to the coverage set $\mathcal{C}$, we might generate either $\delta = 0$ or $\delta = 1$ new target words. For $\delta = 1$, a new target language word is generated using the trigram language model $p(e|e', e'')$. For $\delta = 0$, no new target word is generated, while an additional source sentence position is covered. A modified language model probability $p_\delta(e|e', e'')$ is defined as follows:

$$p_\delta(e|e', e'') = \begin{cases} 1.0 & \text{if } \delta = 0 \\ p(e|e', e'') & \text{if } \delta = 1 \end{cases} .$$

We associate a distribution $p(\delta)$ with the two cases $\delta = 0$ and $\delta = 1$ and set $p(\delta = 1) = 0.7$.
The above auxiliary quantity satisfies the following recursive DP equation:

$$Q_{e'}(e, \mathcal{C}, j) = $$

$$= p(f_j|e) \cdot \max_{\substack{\delta, e'' \\ j' \in \mathcal{C} \setminus \{j\}}} \Big\{ p(j|j', J) \cdot p(\delta) \cdot \cdot p_\delta(e|e', e'') \cdot Q_{e''}(e', \mathcal{C} \setminus \{j\}, j') \Big\}.$$

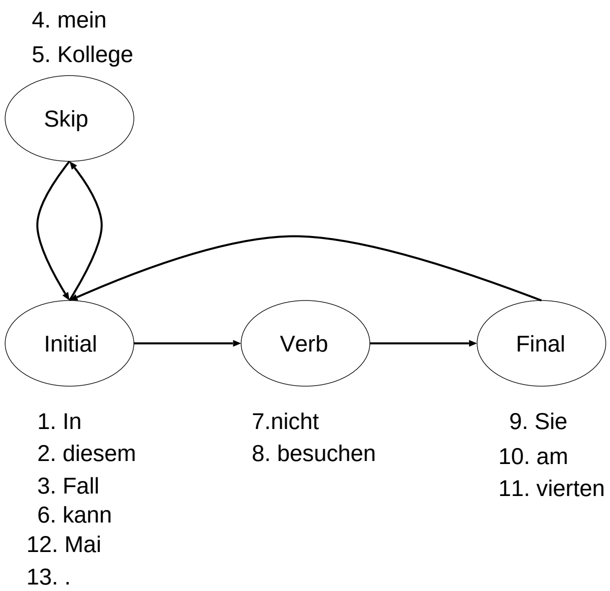

Figure 2: Order in which source positions are visited for the example given in Fig.1.

The DP equation is evaluated recursively for each hypothesis $(e', e, \mathcal{C}, j)$. The resulting algorithm is depicted in Table 1. The complexity of the algorithm is $O(E^3 \cdot J^2 \cdot 2^J)$, where E is the size of the target language vocabulary.

3.1 Word Re-Ordering with Verbgroup Restrictions: Quasi-monotone Search

The above search space is still too large to allow the translation of a medium length input sentence. On the other hand, only very restricted re-orderings are necessary, e.g. for the translation direction from

Table 2: Coverage set hypothesis extensions for the IBM re-ordering.

$No.$	Predecessor coverage set		$\rightarrow$	Successor coverage set
1	$(\{1,\cdots,m\} \setminus \{l\}$	$,l')$	$\rightarrow$	$(\{1,\cdots,m\}, l)$
2	$(\{1,\cdots,m\} \setminus \{l,l_1\}$	$,l')$	$\rightarrow$	$(\{1,\cdots,m\} \setminus \{l_1\}, l)$
3	$(\{1,\cdots,m\} \setminus \{l,l_1,l_2\}$	$,l')$	$\rightarrow$	$(\{1,\cdots,m\} \setminus \{l_1,l_2\}, l)$
4	$(\{1,\cdots,m-1\} \setminus \{l_1,l_2,l_3\}$	$,l')$	$\rightarrow$	$(\{1,\cdots,m\} \setminus \{l_1,l_2,l_3\}, \mathbf{m})$

German to English the monotonicity constraint is violated mainly with respect to the German verbgroup. In German, the verbgroup usually consists of a left and a right verbal brace, whereas in English the words of the verbgroup usually form a sequence of consecutive words. Our new approach, which is called **quasi-monotone** search, processes the source sentence monotonically, while explicitly taking into account the positions of the German verbgroup.

A typical situation is shown in Figure 1. When translating the sentence monotonically from left to right, the translation of the German finite verb 'kann', which is the left verbal brace in this case, is postponed until the German noun phrase 'mein Kollege' is translated, which is the subject of the sentence. Then, the German infinitive 'besuchen' and the negation particle 'nicht' are translated. The translation of one position in the source sentence may be postponed for up to $L = 3$ source positions, and the translation of up to two source positions may be anticipated for at most $R = 10$ source positions. To formalize the approach, we introduce four **verbgroup** states $\mathcal{S}$:

- Initial ($\mathcal{I}$): A contiguous, initial block of source positions is covered.
- Skipped ($\mathcal{K}$): The translation of up to one word may be postponed .
- Verb ($\mathcal{V}$): The translation of up to two words may be anticipated.
- Final ($\mathcal{F}$): The rest of the sentence is processed monotonically taking account of the already covered positions.

While processing the source sentence monotonically, the initial state $\mathcal{I}$ is entered whenever there are no uncovered positions to the left of the rightmost covered position. The sequence of states needed to carry out the word re-ordering example in Fig. 1 is given in Fig. 2. The 13 positions of the source sentence are processed in the order shown. A position is presented by the word at that position. Using these states, we define partial hypothesis extensions, which are of the following type:

$$(\mathcal{S}', \mathcal{C} \setminus \{j\}, j') \rightarrow (\mathcal{S}, \mathcal{C}, j),$$

Not only the coverage set $\mathcal{C}$ and the positions j, j', but also the verbgroup states $\mathcal{S}, \mathcal{S}'$ are taken into account. To be short, we omit the target words e, e' in the formulation of the search hypotheses. There are 13 types of extensions needed to describe the verbgroup re-ordering. The details are given in (Tillmann, 2000). For each extension a new position is added to the coverage set. Covering the first uncovered position in the source sentence, we use the language model probability $p(e|\$,\$)$. Here, \$ is the sentence boundary symbol, which is thought to be at position 0 in the target sentence. The search starts in the hypothesis $(\mathcal{I}, \{\emptyset\}, 0)$. $\{\emptyset\}$ denotes the empty set, where no source sentence position is covered. The following recursive equation is evaluated:

$$\begin{aligned} & Q_{e'}(e, \mathcal{S}, \mathcal{C}, j) = \qquad (2) \\ & = \; p(f_j|e) \cdot \max_{\delta, e''} \Big\{ p(j|j', J) \cdot p(\delta) \cdot p_\delta(e|e', e'') \cdot \\ & \qquad \cdot \max_{\substack{(\mathcal{S}', j') \\ (\mathcal{S}', \mathcal{C}\setminus\{j\}, j') \rightarrow (\mathcal{S}, \mathcal{C}, j) \\ j' \in \mathcal{C}\setminus\{j\}}} Q_{e''}(e', \mathcal{S}', \mathcal{C} \setminus \{j\}, j') \Big\}. \end{aligned}$$

The search ends in the hypotheses $(\mathcal{I}, \{1, \cdots, J\}, j)$. $\{1, \cdots, J\}$ denotes a coverage set including all positions from the starting position 1 to position J and $j \in \{J - L, \cdots, J\}$. The final score is obtained from:

$$\max_{\substack{e, e' \\ j \in \{J-L, \cdots, J\}}} p(\$|e, e') \cdot Q_{e'}(e, \mathcal{I}, \{1, \cdots, J\}, j),$$

where $p(\$|e, e')$ denotes the trigram language model, which predicts the sentence boundary \$ at the end of the target sentence. The complexity of the quasi-monotone search is $O(E^3 \cdot J \cdot (R^2 + L \cdot R))$. The proof is given in (Tillmann, 2000).

3.2 Re-ordering with IBM Style Restrictions

We compare our new approach with the word re-ordering used in the IBM translation approach (Berger et al., 1996). A detailed description of the search procedure used is given in this patent. Source sentence words are aligned with hypothesized target sentence words, where the choice of a new source word, which has not been aligned with a target word yet, is restricted[1]. A procedural definition to restrict

[1]In the approach described in (Berger et al., 1996), a morphological analysis is carried out and word morphemes rather than full-form words are used during the search. Here, we process only full-form words within the translation procedure.

the number of permutations carried out for the word re-ordering is given. During the search process, a partial hypothesis is extended by choosing a source sentence position, which has not been aligned with a target sentence position yet. Only one of the first n positions which are not already aligned in a partial hypothesis may be chosen, where n is set to 4. The restriction can be expressed in terms of the number of uncovered source sentence positions to the left of the rightmost position m in the coverage set. This number must be less than or equal to $n-1$. Otherwise for the predecessor search hypothesis, we would have chosen a position that would not have been among the first n uncovered positions.
Ignoring the identity of the target language words e and e', the possible partial hypothesis extensions due to the IBM restrictions are shown in Table 2. In general, $m, l, l' \neq \{l_1, l_2, l_3\}$ and in line umber 3 and 4, l' must be chosen not to violate the above re-ordering restriction. Note that in line 4 the last visited position for the successor hypothesis must be m. Otherwise , there will be four uncovered positions for the predecessor hypothesis violating the restriction. A dynamic programming recursion similar to the one in Eq. 2 is evaluated. In this case, we have no finite-state restrictions for the search space. The search starts in hypothesis $(\{\emptyset\}, 0)$ and ends in the hypotheses $(\{1, \cdots, J\}, j)$, with $j \in \{1, \cdots, J\}$. This approach leads to a search procedure with complexity $O(E^3 \cdot J^4)$. The proof is given in (Tillmann, 2000).

4 Experimental Results

4.1 The Task and the Corpus

We have tested the translation system on the Verbmobil task (Wahlster 1993). The Verbmobil task is an appointment scheduling task. Two subjects are each given a calendar and they are asked to schedule a meeting. The translation direction is from German to English. A summary of the corpus used in the experiments is given in Table 3. The perplexity for the trigram language model used is 26.5. Although the ultimate goal of the Verbmobil project is the translation of *spoken* language, the input used for the translation experiments reported on in this paper is the (more or less) correct orthographic transcription of the spoken sentences. Thus, the effects of spontaneous speech are present in the corpus, e.g. the syntactic structure of the sentence is rather less restricted, however the effect of speech recognition errors is not covered.
For the experiments, we use a simple preprocessing step. German city names are replaced by category markers. The translation search is carried out with the category markers and the city names are resubstituted into the target sentence as a postprocessing step.

Table 3: Training and test conditions for the Verbmobil task (*number of words without punctuation marks).

		German	English
Training:	Sentences	58 073	
	Words	519 523	549 921
	Words*	418 979	453 632
Vocabulary	Size	7939	4648
	Singletons	3454	1699
Test-147:	Sentences	147	
	Words	1 968	2 173
	Perplexity	–	26.5

Table 4: Multi-reference word error rate (mWER) and subjective sentence error rate (SSER) for three different search procedures.

Search Method	CPU time [sec]	mWER [%]	SSER [%]
MonS	0.9	42.0	30.5
QmS	10.6	34.4	23.8
IbmS	28.6	38.2	26.2

4.2 Performance Measures

The following two error criteria are used in our experiments:

- mWER: multi-reference WER:
  We use the Levenshtein distance between the automatic translation and several reference translations as a measure of the translation errors. On average, 6 reference translations per automatic translation are available. The Levenshtein distance between the automatic translation and each of the reference translations is computed, and the minimum Levenshtein distance is taken. This measure has the advantage of being completely automatic.

- SSER: subjective sentence error rate:
  For a more detailed analysis, the translations are judged by a human test person. For the error counts, a range from 0.0 to 1.0 is used. An error count of 0.0 is assigned to a perfect translation, and an error count of 1.0 is assigned to a semantically and syntactically wrong translation.

4.3 Translation Experiments

For the translation experiments, Eq. 2 is recursively evaluated. We apply a beam search concept as in speech recognition. However there is no global pruning. Search hypotheses are processed separately according to their coverage set $\mathcal{C}$. The best scored hypothesis for each coverage set is computed:

$$Q_{Beam}(\mathcal{C}) \quad = \quad \max_{e,e',\mathcal{S},j} Q_{e'}(e,\mathcal{S},\mathcal{C},j)$$

The hypothesis $(e', e, \mathcal{S}, \mathcal{C}, j)$ is pruned if:

$$Q_{e'}(e,\mathcal{S},\mathcal{C},j) \; < \; t_0 \cdot Q_{Beam}(\mathcal{C}),$$

where t_0 is a threshold to control the number of surviving hypotheses. Additionally, for a given coverage set, at most 250 different hypotheses are kept during the search process, and the number of different words to be hypothesized by a source word is limited. For each source word f, the list of its possible translations e is sorted according to $p(f|e) \cdot p_{uni}(e)$, where $p_{uni}(e)$ is the unigram probability of the English word e. It is sufficient to consider only the best 50 words.

We show translation results for three approaches: the monotone search (**MonS**), where no word reordering is allowed (Tillmann, 1997), the quasi-monotone search (**QmS**) as presented in this paper and the IBM style (**IbmS**) search as described in Section 3.2.

Table 4 shows translation results for the three approaches. The computing time is given in terms of CPU time per sentence (on a 450-MHz Pentium-III-PC). Here, the pruning threshold $t_0 = 10.0$ is used. Translation errors are reported in terms of multi-reference word error rate (mWER) and subjective sentence error rate (SSER). The monotone search performs worst in terms of both error rates mWER and SSER. The computing time is low, since no reordering is carried out. The quasi-monotone search performs best in terms of both error rates mWER and SSER. Additionally, it works about 3 times as fast as the IBM style search. For our demonstration system, we typically use the pruning threshold $t_0 = 5.0$ to speed up the search by a factor 5 while allowing for a small degradation in translation accuracy.

The effect of the pruning threshold t_0 is shown in Table 5. The computing time, the number of search errors, and the multi-reference WER (mWER) are shown as a function of t_0. The negative logarithm of t_0 is reported. The translation scores for the hypotheses generated with different threshold values t_0 are compared to the translation scores obtained with a conservatively large threshold $t_0 = 10.0$. For each test series, we count the number of sentences whose score is worse than the corresponding score of the test series with the conservatively large threshold $t_0 = 10.0$, and this number is reported as the number of search errors. Depending on the threshold t_0, the search algorithm may miss the globally optimal path which typically results in additional translation errors. Decreasing the threshold results in higher mWER due to additional search errors.

Table 5: Effect of the beam threshold on the number of search errors (147 sentences).

Search Method	t_0	CPU time [sec]	#search error	mWER [%]
QmS	0.0	0.07	108	42.6
	1.0	0.13	85	37.8
	2.5	0.35	44	36.6
	5.0	1.92	4	34.6
	10.0	10.6	0	34.5
IbmS	0.0	0.14	108	43.4
	1.0	0.3	84	39.5
	2.5	0.8	45	39.1
	5.0	4.99	7	38.3
	10.0	28.52	0	38.2

Table 6 shows example translations obtained by the three different approaches. Again, the monotone search performs worst. In the second and third translation examples, the IbmS word re-ordering performs worse than the QmS word re-ordering, since it can not take properly into account the word re-ordering due to the German verbgroup. The German finite verbs 'bin' (second example) and 'könnten' (third example) are too far away from the personal pronouns 'ich' and 'Sie' (6 respectively 5 source sentence positions). In the last example, the less restrictive IbmS word re-ordering leads to a better translation, although the QmS translation is still acceptable.

5 Conclusion

In this paper, we have presented a new, efficient DP-based search procedure for statistical machine translation. The approach assumes that the word reordering is restricted to a few positions in the source sentence. The approach has been successfully tested on the 8 000-word Verbmobil task. Future extensions of the system might include: 1) An extended translation model, where we use more context to predict a source word. 2) An improved language model, which takes into account syntactic structure, e.g. to ensure that a proper English verbgroup is generated. 3) A tight coupling with the speech recognizer output.

Acknowledgements

This work has been supported as part of the Verbmobil project (contract number 01 IV 601 A) by the German Federal Ministry of Education, Science, Research and Technology and as part of the Eutrans project (ESPRIT project number 30268) by the European Community.

Table 6: Example Translations for the Verbmobil task.

Input:	Ja , wunderbar . Können wir machen .
MonS:	Yes, wonderful. Can we do .
QmS:	Yes, wonderful. We can do that .
IbmS:	Yes, wonderful. We can do that .
Input:	Das ist zu knapp , weil ich ab dem dritten in Kaiserslautern bin . Genaugenommen nur am dritten . Wie wäre es denn am ähm Samstag , dem zehnten Februar ?
MonS:	That is too tight , because I from the third in Kaiserslautern . In fact only on the third . How about ähm Saturday , the tenth of February ?
QmS:	That is too tight , because I am from the third in Kaiserslautern . In fact only on the third . Ähm how about Saturday , February the tenth ?
IbmS:	That is too tight , from the third because I will be in Kaiserslautern . In fact only on the third . Ähm how about Saturday , February the tenth ?
Input:	Wenn Sie dann noch den siebzehnten könnten , wäre das toll , ja .
MonS:	If you then also the seventeenth could , would be the great , yes .
QmS:	If you could then also the seventeenth , that would be great , yes .
IbmS:	Then if you could even take seventeenth , that would be great , yes .
Input:	Ja , das kommt mir sehr gelegen . Machen wir es dann am besten so .
MonS:	Yes , that suits me perfectly . Do we should best like that .
QmS:	Yes , that suits me fine . We do it like that then best .
IbmS:	Yes , that suits me fine . We should best do it like that .

References

A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, A. S. Kehler, 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, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, vol. 19, no. 2, pp. 263–311.

M. Held, R. M. Karp. 1962. A Dynamic Programming Approach to Sequencing Problems. *J.SIAM*, vol. 10, no. 1, pp. 196–210.

H. Ney, S. Niessen, F. J. Och, H. Sawaf, C. Tillmann, S. Vogel. 2000. Algorithms for Statistical Translation of Spoken Language. *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 24–36.

F. J. Och, C. Tillmann, H. Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP99)*, pp. 20–28, University of Maryland, College Park, MD, USA, June.

F. J. Och and Ney, H. 2000. A comparison of alignment models for statistical machine translation. In *Proceedings of COLING 2000: The 18th International Conference on Computational Linguistics*, Saarbrücken, Germany, July-August.

C. Tillmann. 2000. Complexity of the Different Word Re-ordering Approaches. The document can be found under the URL `http://www-i6.Informatik.RWTH-Aachen.de/Colleagues/tilli/`. Aachen University of Technology, Aachen, Germany, June.

C. Tillmann, S. Vogel, H. Ney and A. Zubiaga. 1997. A DP based Search Using Monotone Alignments in Statistical Translation. In *Proc. of the 35th Annual Conf. of the Association for Computational Linguistics*, pp. 289–296, Madrid, Spain, July.

W. Wahlster. 1993. Verbmobil: Translation of Face-to-Face Dialogs. *MT Summit IV*, pp. 127–135, Kobe, Japan.