Efficient Statistical Machine Translation with Constrained Reordering

Evgeny Matusov, Stephan Kanthak, and Hermann Ney

Lehrstuhl für Informatik VI, Computer Science Department,
RWTH Aachen University
D-52056 Aachen, Germany
{matusov,kanthak,ney}@informatik.rwth-aachen.de.

Abstract. This paper describes how word alignment information makes machine translation more efficient. Following a statistical approach based on finite-state transducers, we perform reordering of source sentences in training using automatic word alignments and estimate a phrase-based translation model. Using this model, we translate monotonically taking a permutation graph as input. The permutation graph is constrained using an efficient and flexible reordering framework. We then propose to automatically identify source word sequences which should always be translated monotonically and keep the word order of these sequences in search. This allows us to obtain fast good-quality translations. We present competitive experimental results on the Verbmobil German-to-English and BTEC Chinese-to-English translation tasks.

1 Introduction

Word reordering is of crucial importance for machine translation. Most of the phrase-based statistical approaches like the Alignment Template system of (Och et al., 2004) rely on reorderings which are implicitly memorized with each pair of source and target phrases in training. Additional reorderings on phrase level are fully integrated into the decoding process, which increases the complexity of the system and makes it hard to modify.

Other statistical approaches make use of the efficient search representation with weighted finite-state transducers (WFSTs). Many of these approaches use joint probabilities of the source and the target language string. The automated transducer inference techniques OMEGA (Vilar, 2000) and GIATI (Casacuberta et al., 2004) estimate phrase-based models, but capture reordering only implicitly in bilingual corpus representations. This leads to a strong degradation of translation quality when translating into a language with a completely different word order. In (Bangalore et al., 2000) weighted reordering has been applied to target sentences. In order to reduce the computational complexity, this approach considers only a set of plausible reorderings seen on training data.

In this paper, we follow a phrase-based joint-probability WFST translation approach, in which source sentence reordering is applied on word level, both in training and for translation. This is a novel approach inspired by the work of (Knight et al., 1998) and (Kumar et al., 2003). In this approach, a reordering graph is computed on-demand and taken as input for monotonic translation. The approach is modular and allows easy introduction of different reordering constraints and probabilistic dependencies. Here, we extend this approach by introducing additional restrictions on reorderings. We describe our efficient finite-state implementations of IBM (Berger et al., 1996), inverse IBM and local reordering constraints. Furthermore, we apply these constraints in the search, but keep the word order within source phrases which were consistently aligned monotonically in training.

In the next section we review the general theory of our translation system based on weighted finite-state transducers and describe the use of word alignments for reordering in training. We then discuss three modeling techniques which use alignment information to establish connections between source and target words. These connections are needed in order to estimate the joint translation probability. Section 3 describes the on-demand computable

framework for permutation models and the various types of the reordering constraints that are applied in the search. In Section 4 we propose how the reordering process can be further constrained by keeping the order of monotonic source sequences. We conclude the paper with experimental results, which show the advantages of these constraints on two translation tasks.

2 Basics of the Translation System

2.1 Bayes Decision Rule

In statistical machine translation, we are looking for a target language sentence e_1^I which translates a source sentence f_1^J. We formulate the Bayes decision rule for maximizing the posterior probability:

$$\begin{aligned}
\hat{e_1^I} &= \operatorname*{argmax}_{I,e_1^I} Pr(e_1^I|f_1^J) \\
&= \operatorname*{argmax}_{I,e_1^I} Pr(f_1^J, e_1^I) \\
&= \operatorname*{argmax}_{I,e_1^I} \sum_{\mathcal{A}} Pr(\mathcal{A}) \cdot Pr(f_1^J, e_1^I|\mathcal{A}) \\
&\cong \operatorname*{argmax}_{I,e_1^I} \max_{\mathcal{A}} Pr(\mathcal{A}) \cdot Pr(f_1^J, e_1^I|\mathcal{A})
\end{aligned}$$

Here, the posterior probability $Pr(e_1^I|f_1^J)$ is rewritten as a *joint* probability of the input and the output sentence. The stochastic finite-state transducer approach allows for convenient modeling of joint probabilities. We also assume that we have word level alignments $\mathcal{A}$ of all sentence pairs from a bilingual training corpus and introduce such alignments as a hidden variable.

2.2 Word Alignments

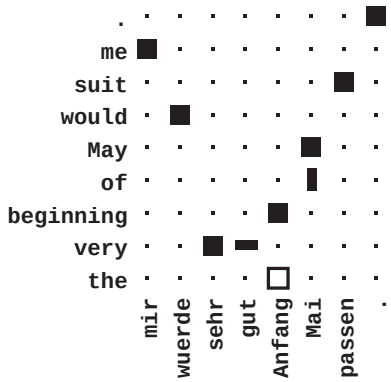
The statistical word alignments are used in two ways. First, we reorder the words in each training source sentence based on an alignment which is a function of source words and naturally defines their permutation (Figure 1). This allows to train a monotonic translation model.

Next, the goal is to establish connections between the reordered source and the target words in order to reliably estimate the joint translation probability with statistical language modeling techniques. To this end, most of the WFST approaches aim at a "bilanguage" representation of each pair of sentences in the training corpus with K bilingual phrases $(\tilde{f}_k, \tilde{e}_k)$, $k = 1, ..., K$ of varying length. All of these methods do not require, but work especially well with fully monotonic alignments. Here, we discuss the most common techniques.

The representation used by e. g. (Bangalore et al., 2000) allows source and target phrases $\tilde{f}$ and $\tilde{e}$ to have the length of either 0 or 1. This means that each pair of training sentences is written with bilingual tuples (f, e) where either f or e can be a normal word or an "empty word" which we denote with \$. To create a corpus of such bilingual pairs, a *one-to-one* alignment is used. The number of bilingual phrases K can vary from $\max(I, J)$ to $(I+J)$. Whereas the vocabulary size of the corpus in this representation is relatively limited, m-gram models with a long history m have to be built to capture enough phrasal context. Also, the complexity of the WFST search increases, since epsilon arcs have to be used in order to hypothesize non-aligned target words.

In the representation of (Casacuberta et al., 2004) $\tilde{f}$ is one real source word only, and $\tilde{e}$ is a contiguous target phrase of 0 or more words. This representation arises from *one-to-many* alignments which are functions of target words. The advantage of this representation is that the search effort is proportional to the length of the source sentence. However, the vocabulary size of the "bilanguage" increases. This may result in data sparsity problems, which at least partially can be solved with smoothing techniques.

Finally, (de Gispert et al., 2002) describe bilingual X-grams $(\tilde{f}, \tilde{e})$ without restrictions on the length of source or target phrase. This representation can be derived from a general alignment with *many-to-many* connections. The drawback of this representation is the enormous vocabulary size which may not allow for reliable estimation of the translation probability. Another disadvantage is the inability to translate individual words in $\tilde{f}$, if e. g. they do not appear in the training corpus in another context. In our opinion, however, in at least two




```
   sehr gut Anfang Mai wuerde passen mir .
   the very beginning of May would suit me .

A) $|the sehr|very gut|$ Anfang|beginning $|of
   Mai|May wuerde|would passen|suit mir|me .|.

B) sehr|the_very gut|$ Anfang|beginning
   Mai|of_May wuerde|would passen|suit mir|me .|.

C) sehr_gut|the_very Anfang|beginning
   Mai|of_May wuerde|would passen|suit mir|me .|.
```

Figure 1. An example of alignment, source sentence reordering, monotonization and some alternative bilingual corpus representations. Alignment connections: ■ used for reordering and all representations; ▬ used for reordering and representation (C); ▮ used for representations (B) and (C); □ ignored due to the monotonicity requirements.

cases it may be reasonable to include some phrases $\tilde{f}$ with length > 1. The first case is when several source words are always translated with one target word (e. g. translating an English noun phrase into a German compound). The second case usually involves non-literal phrase-to-phrase translations, when translating individual source words does not convey the meaning of the source phrase.

An example of the source sentence reordering, as well as of the three described bilingual corpus representations, labeled with (A),(B), and (C), respectively, is given in Figure 1.

In our approach, we can avoid various heuristics and learn these and other types of corpus representations by using a flexible alignment framework presented in (Matusov et al., 2004). Following this work, we efficiently compute optimal, minimum-cost alignments which satisfy certain constraints. The constraints may include the requirement for each word to be aligned at least once, functional form or full monotonicity. Local alignment costs between a source word f_j and a target word e_i are estimated statistically using state occupation probabilities of the HMM and IBM-4 models as trained by the GIZA++ toolkit (Och et al., 2003).

2.3 Optimization Criterion

Using one of the corpus representations $(\tilde{f}, \tilde{e})$ via a certain (constrained) alignment A, we rewrite the joint translation probability in the decision rule as follows:

$$\begin{aligned}
\hat{e}_1^I &= \operatorname*{argmax}_{I,e_1^I} \max_A Pr(A) \cdot Pr(f_1^J, e_1^I|A) \\
&= \operatorname*{argmax}_{\tilde{e}_1^K} \max_{A,K} Pr(A) \cdot Pr(\tilde{f}_1^K, \tilde{e}_1^K|A, K) \\
&\cong \operatorname*{argmax}_{\tilde{e}_1^K,A,K} \prod_{k=1}^{K} Pr(\tilde{f}_k, \tilde{e}_k|\tilde{f}_1^{k-1}, \tilde{e}_1^{k-1}, A, K) \\
&\cong \operatorname*{argmax}_{\tilde{e}_1^K,A,K} \prod_{k=1}^{K} p(\tilde{f}_k, \tilde{e}_k|\tilde{f}_{k-m}^{k-1}, \tilde{e}_{k-m}^{k-1}, A, K)
\end{aligned}$$

In other words: the translation problem is mapped to the problem of estimating an m-gram language model over a learned set of bilingual tuples $(\tilde{f}_k, \tilde{e}_k)$. Mapping the bilingual language model to a WFST T is canonical.

3 Reordering in Search

Since we chose to reorder source sentences in training and translate monotonically, we can properly translate only sentences which have the word order of the target language. To overcome this obstacle, the input sentence has to be permuted, and the translation model will then select the best path through the permutation graph in a global decision process.

When searching the best translation $\tilde{e}_1^K$ for a given source sentence f_1^J, we firstly represent this input sentence as a linear automaton with word-labeled arcs (see top of Figure 3). We then compute permutations of this automaton as described

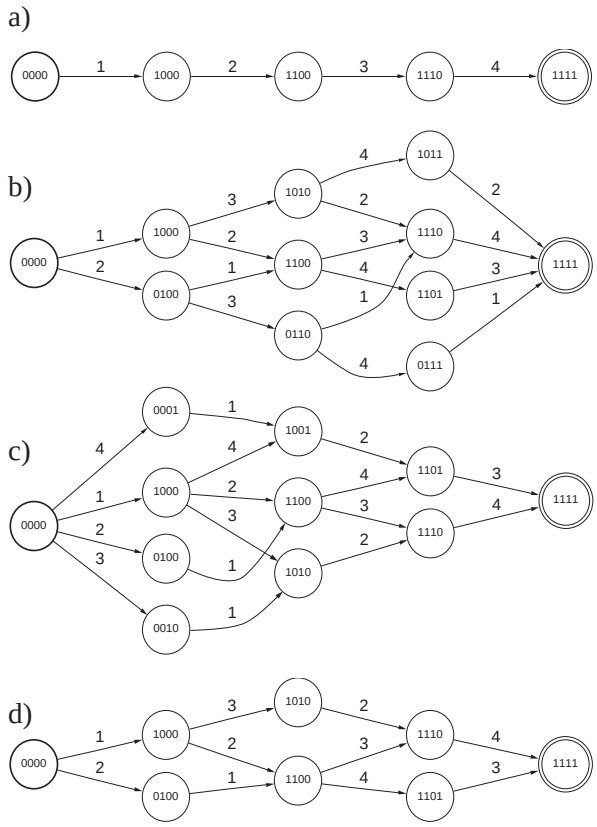


Figure 2. Permutations of a) positions $j = 1, 2, 3, 4$ of a source sentence $f_1 f_2 f_3 f_4$ using a window size of 2 for b) IBM constraints, c) inverse IBM constraints and d) local constraints.

in (Knight et al., 1998). The overall search problem can be rewritten using finite-state terminology (Kanthak et al., 2004):

$$\hat{e}_1^I = \texttt{project-output}(\texttt{best}(\texttt{permute}(f_1^J) \circ T))$$

This implementation of the search problem with weighted finite-state transducers is very efficient. However, permuting an input sequence of J symbols results in $J!$ possible permutations, i. e. in exponential complexity. Therefore, we compute a constrained permutation automaton on-demand while optionally applying beam pruning in the search.

For on-demand computation of an automaton we specify a state description and an algorithm that calculates all outgoing arcs of a state from the state description. In our case, each state represents a permutation of a subset of the source words f_1^J, which are already translated. This can be described by a bit vector b_1^J. Each bit of the state bit vector corresponds to an arc of the linear input automaton and is set to one if the arc has been used on any path from the initial to the current state. The bit vectors of two states connected by an arc differ only in a single bit. Note that bit vectors elegantly solve the problem of recombining paths in the automaton as states with the same bit vectors can be merged. As a result, a fully minimized permutation automaton has only a single initial and final state.

Even with on-demand computation, complexity using full permutations is unmanageable for long sentences. We further reduce complexity by additionally limiting permutations with the constraints, which we describe in the following. For all of these constraints, we use implementations with bit vector state descriptions to compute constrained permutation graphs on-demand. Refer to Figure 2 for their visualizations.

The IBM reordering constraints are well-known in the field of machine translation and were first described in (Berger et al., 1996). The idea behind these constraints is to deviate from monotonic translation by postponing translations of a limited number of words. More specifically, at each state we can translate any of the *first* l yet uncovered word positions. For consistency we associate *window size* with the parameter l.

For some language pairs, it is beneficial to translate some words at the end of the sentence first and to translate the rest of the sentence nearly monotonically. Following this idea we can define the *inverse IBM constraints*. Let j be the first uncovered position. We can choose any position for translation, unless $l - 1$ words on positions $j' > j$ have been translated. If this is the case we must translate the word in position j.

For some language pairs, e.g. Italian – English, words are moved only a few positions to the left or right. The IBM constraints provide too many alternative permutations to chose from as each word can be moved to the end of the sentence. A solution that allows only for local permutations and therefore has very low complexity is given by the following permutation rule: the next word for translation comes

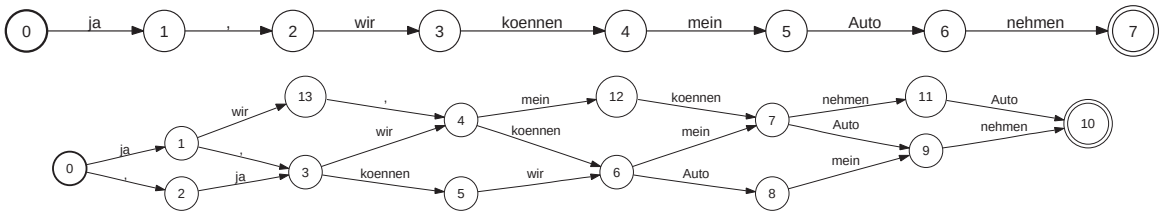


Figure 3. An example of local constraints with window size of 2.

from the window of l positions[1] counting from the first yet uncovered position. Note, that the local constraints define a true subset of the permutations defined by the IBM constraints. Figure 3 illustrates these most restrictive, but efficient constraints with the window size of 2 when permuting the German sentence "ja, wir können mein Auto nehmen".

We also introduce weights to the constraints and normally give higher probability to the arcs of the monotonic path through the reordering graph, while penalizing the non-monotonic ones.

4 Monotonic Sequences

Even with constrained reordering, the search space, especially for long sentences, may become too large to handle. However, many paths in the reordering graph are not relevant for translation and may even be harmful for the performance. Usually, each input source sentence can be viewed as several sequences of $n \geq 1$ words, each of which should be translated monotonically.

We propose to identify such sequences in training and forbid permutations which change the word order within such sequences, or break them up. To this end, we collect statistics over the training corpus by considering alignments which are functions of source words. We extract consecutive source phrases of various length (≤ 10), which were consistently aligned with some target words in a monotonic way.

When translating a source sentence, we search for monotonic sequences observed in training and perform longest match. We then concatenate all the words in the found monotonic sequences and use them to label only one arc in the linear automaton (see e. g. top of Figure 4). When overlapping matches exist, we unite the matched sequences and thus are able to identify longer monotonic sequences not observed in training.

We permute the transformed linear automaton under some constraints using on-demand computation. Next, we make the reverse transformation and replace each arc in the reordering graph which is labeled with n words by n single-word arcs. This allows us to apply the bilingual m-gram language model transducer on the original lexical entries and make use of its generalization capability. All of these steps are efficiently realized at runtime with generic composition operations. The resulting permutation graph is shown in Figure 4. Note that it is significantly more compact than the corresponding graph in Figure 3 and contains the most plausible reorderings only. In particular, the movements of the verb "nehmen" are not restricted, which makes it possible for the system to choose the sequence of arcs "können nehmen" for correct phrasal translation with "can take".

It is also possible to generalize from the monotonic sequences in training by matching corresponding sequences of word classes or part-of-speech tags. Another application of the presented technique would be to explicitly forbid reorderings of word sequences which must be a-priori translated monotonically, like sequences of digits, time and date expressions, multi-word names, spelled letters, etc.. Such restrictions are especially important for subjective user appreciation of the system's performance.

5 Experimental Results

5.1 Corpus Statistics

The translation experiments were carried out on the *Basic Travel Expression Corpus* (BTEC), a multilingual speech corpus which contains tourism-related sentences usually found in travel phrase books. We tested our system on the Chinese-

[1] both covered and uncovered

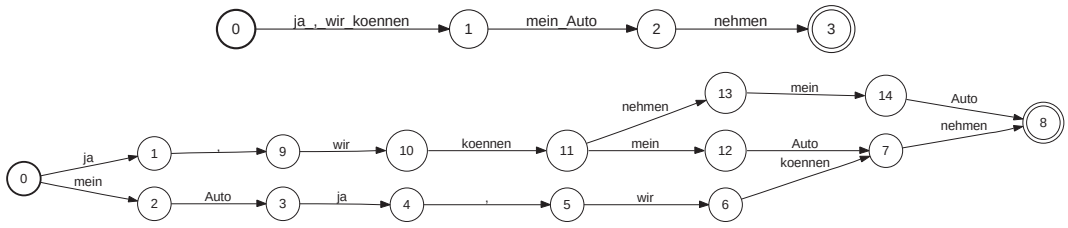


Figure 4. Reordering with local constraints and window size of 2 and non-reordered monotonic sequences.

		Chinese	English
Train	sentences	20 000	
	words	182 904	160 523
	singletons	3 525	2 948
	vocabulary	7 643	6 982
Test	sentences	506	
	words	3 515	3 595

Table 1. Statistics of the Basic Travel Expression corpus.

		German	English
Train	Sentences	58 073	
	Words	519 523	549 921
	Vocabulary	7 939	4 672
	Singletons	3 453	1 698
Lexicon	Entries	12 779	
Test	Sentences	251	
	Words	2 628	2 871

Table 2. Statistics of the Verbmobil corpus.

to-English Supplied Task, the corpus for which was provided during the International Workshop on Spoken Language Translation (IWSLT 2004) (Akiba et al., 2004). The corpus statistics for the BTEC corpus are given in Table 1. We evaluate the impact of the proposed reordering restrictions on the CSTAR 2003 test set with 506 Chinese sentences and 16 reference translations.

We also present results on 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 statistics of this corpus.

5.2 Evaluation Criteria

For the automatic evaluation, we used the word error rate (WER), position-independent word error rate (PER), and the BLEU score (Papineni et al., 2002). This score measures accuracy, i. e. larger scores are better. The three measures were computed with respect to *m*ultiple reference translations, when they were available. To indicate this, we will label the error rate acronyms with an m. On the Chinese-to-English BTEC task, both training and evaluation were performed using corpora and references in lowercase and without punctuation marks.

5.3 Experiments

As described in Sec. 2.2, we reordered the source sentences in training. We then created a bilingual corpus of tuples $(f_j, \tilde{e}_j)$ (i. e. representation (B) in Figure 1) based on a fully monotonic alignment that is a function of target words. Using this corpus, we estimated a smoothed m-gram language model[2] and represented it as a finite-state transducer.

When translating, we applied moderate beam pruning to the search graph only when necessary. This allowed for reasonable translation times and memory consumption without a significant negative impact on performance. In baseline experiments, we did not reorder source sentences in the search. In all other experiments where constrained reordering was permitted, we obtained most optimal results when we restricted reordering in matched word sequences which had been monotonically aligned in training more than 50 % of the time. With this setting, the average number of arcs in a linear automaton representation of a sentence decreased from 7 to about 5 for the BTEC test set, and dramatically from more than 10 to 6 for the Verbmobil test set.

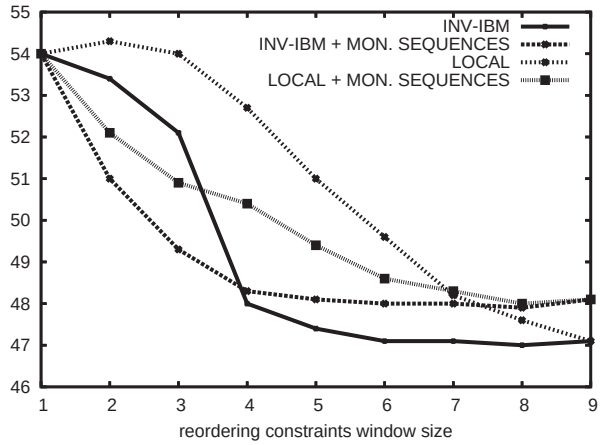


Figure 5. Word error rate [%] as a function of the reordering window size for different reordering constraints: Chinese-to-English translation.

	mWER [%]	mPER [%]	BLEU [%]	speed [w/s]	mem [MB]
baseline*	54.0	42.3	23.0	110	28
4-inv-ibm	48.0	39.4	33.2	0.5	402
3-inv-ibm$^\diamondsuit$	49.1	40.3	30.1	7	94
5-local$^\diamondsuit$	49.4	40.2	31.2	56	62

Table 3. Translation quality and efficiency on the BTEC task, development corpus (*:full search; $^\diamondsuit$: with fixed word order in monotonic sequences).

5.3.1 Chinese-to-English Translation

Word order in Chinese and English is somewhat similar. However, a few word reorderings over quite large distances may be necessary. This is especially true in case of questions, in which question words like "where" and "when" are placed, unlike in English, at the end of a sentence. Based on these observations, we expected that identifying monotonic sequences will result in faster and better translations under reordering constraints with small window sizes.

The best translation results for this task were achieved under inverse IBM reordering constraints with window size ≥ 4. Figure 5 shows that using monotonic sequences in which the words are not permuted in search, we can achieve similar performance with window size 3. The local constraints generally perform well on this task only for very large window sizes ≥ 9. By keeping the word order in monotonic sequences, we are able to reach similar performance with a window size of 5 or 6. Table 3 presents all error measures, as well as time and memory usage for three configurations with similar word error rate. Keeping the word order in monotonic sequences fixed, we observed dramatic improvements in translation speed from 0.5 to 7, or even to 56 words per second[3] without a large degradation of the performance.

The increase in the word error rate for larger window sizes with the proposed restrictions can be explained by insufficient alignment quality. In some alignments in training, source word sequences were incorrectly aligned monotonically. Their permutation may be useful, but is not performed in translation process.

5.3.2 German-to-English Translation

German language differs in word order from English mainly in the position of verbs and verb prefixes, which often appear at the end of a sentence. Reordering is very important to achieve good translation performance.

The Alignment Template system of (Och et al., 2004) performs phrasal reordering using a complicated graph search algorithm with extensive pruning and heuristic functions for rest cost estimation. It also incorporates several features like the lexicon scores, word penalty, etc., the scaling factors for which have to be optimized. In contrary, our system uses only the translation model score and limited computational resources, so that pruning is often not necessary and search errors can be avoided altogether. Nevertheless, using weighted constrained reorderings in search, we can report competitive translation results.

For three different types of reordering constraints, the window size and probability for the monotonic path were optimized on a development set. The best word error rate on the test set is achieved under IBM constraints with a window size of 4 (see Table 4). Pruning is necessary, and the translation speed is 8 words per second. Using inverse IBM constraints, we are able to get the lowest position-independent error rate of 26.5 % reported in (Och et al., 2004). Here we perform full

[2] $m = 4$ on the BTEC task, $m = 3$ on the Verbmobil task.

[3] 2 x Pentium III 600MHz, 1GB RAM.

	mWER [%]	PER [%]	BLEU [%]	speed [w/s]	mem [MB]
baseline	41.5	29.1	40.6	170	28
3-inv-ibm	37.5	26.5	50.5	2	80
4-ibm*	36.2	27.4	49.1	8	62
2-inv-ibm	36.9	26.9	50.3	13	37
3-local$^\diamond$	36.3	27.3	49.9	35	53

Table 4. Translation quality and efficiency on the Verbmobil task (*: with beam pruning; $^\diamond$: with fixed word order in monotonic sequences).

search and translate at a speed of 2 words per second. At the same time, when we keep the word order in the monotonic sequences fixed, we can produce translations of almost the same quality using the efficient local constraints at the rate of 35 words per second. Thus, the efficiency of the translation increases without significant loss in performance. Fast translations are quite important in the applications similar to the original Verbmobil project – speech-to-speech dialogue translation.

6 Conclusion

In this paper, we described a novel extension to the reordering framework which performs source sentence reordering on word level. We employed a monotonic phrase-based translation system that takes a reordering graph as input. Based on statistics for monotonically aligned source word sequences in training, we identified source phrases in the input sentences, the word order in which should be kept fixed. Using an efficient finite-state implementation, we included the modeling of such phrases into the framework which realizes constrained, weighted, on-demand computable permutations. We showed that this new component significantly improves the efficiency of the search, while allowing quality translations into a language with different word order. We achieved competitive results on Chinese-to-English and German-to-English tasks. In the future, we would like to explore more sophisticated probability distributions for the reordering alternatives.

Acknowledgement

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG) under the project "Statistische Textübersetzung" (Ne572/5) and by the European Union under the integrated project TC-STAR – Technology and Corpora for Speech to Speech Translation (IST-2002-FP6-506738).

7 References

Akiba, Y., Federico, M., Kando, N., Nakaiwa, H., Paul, M., and Tsujii, J. (2004). *Overview of the IWSLT04 Evaluation Campaign*. Proc. Int. Workshop on Spoken Language Translation, pp. 1–12, Kyoto, Japan.

Bangalore, S. and Riccardi, G. (2000). *Stochastic Finite-State Models for Spoken Language Machine Translation*. Proc. Workshop on Embedded Machine Translation Systems, pp. 52–59.

Berger, A. L., Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Gillett, J. R., Kehler, A. S., and Mercer, R. L. (1996). *Language Translation Apparatus and Method of Using Context-based Translation Models*. United States Patent 5510981.

Casacuberta, F. and Vidal, E. (2004). Machine Translation with Inferred Stochastic Finite-State Transducers. Computational Linguistics, vol. 30(2):205-225.

de Gispert, A. and Mariño, J. (2002). *Using X-grams for Speech-to-Speech Translation*. Proc. of the 7th Int. Conf. on Spoken Language Processing, ICSLP'02.

Kanthak, S. and Ney, H. (2004). *FSA: An Efficient and Flexible C++ Toolkit for Finite State Automata using On-demand Computation*. Proc. 42nd Annual Meeting of the ACL, pp. 510–517, Barcelona, Spain.

Knight, K. and Al-Onaizan, Y. (1998). *Translation with Finite-State Devices*. Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1529, pp. 421–437.

Kumar, S. and Byrne, W. (2003). *A Weighted Finite State Transducer Implementation of the Alignment Template Model for Statistical Machine Translation*. Proc. Human Language Technology Conf. NAACL, pp. 142–149, Edmonton, Canada.

Matusov, E., Zens, R., and Ney, H. (2004). *Symmetric Word Alignments for Statistical Machine Translation*. Proc. 20th Int. Conf. on Computational Linguistics, pp. 219–225, Geneva, Switzerland.

Och, F. J. and Ney, H. (2003). *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics, vol. 29, number 1, pp. 19–51.

Och, F. J. and Ney, H. (2004). *The Alignment Template Approach to Statistical Machine Translation*. Computational Linguistics, vol. 30(4):417-449.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proc. 40th Annual Meeting of the ACL, Philadelphia, PA, pp. 311–318.

Vilar, J. M. (2000). *Improve the Learning of Sub-sequential Transducers by Using Alignments and Dictionaries*. Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1891, pp. 298–312.

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