The Alignment Template Approach to Statistical Machine Translation

Franz Josef Och*
Google

Hermann Ney†
RWTH Aachen

A phrase-based statistical machine translation approach — the alignment template approach — is described. This translation approach allows for general many-to-many relations between words. Thereby, the context of words is taken into account in the translation model, and local changes in word order from source to target language can be learned explicitly. The model is described using a log-linear modeling approach, which is a generalization of the often used source–channel approach. Thereby, the model is easier to extend than classical statistical machine translation systems. We describe in detail the process for learning phrasal translations, the feature functions used, and the search algorithm. The evaluation of this approach is performed on three different tasks. For the German–English speech VERBMOBIL *task, we analyze the effect of various system components. On the French–English Canadian* HANSARDS *task, the alignment template system obtains significantly better results than a single-word-based translation model. In the Chinese–English 2002 National Institute of Standards and Technology (NIST) machine translation evaluation it yields statistically significantly better NIST scores than all competing research and commercial translation systems.*

1. Introduction

Machine translation (MT) is a hard problem, because natural languages are highly complex, many words have various meanings and different possible translations, sentences might have various readings, and the relationships between linguistic entities are often vague. In addition, it is sometimes necessary to take world knowledge into account. The number of relevant dependencies is much too large and those dependencies are too complex to take them all into account in a machine translation system. Given these boundary conditions, a machine translation system has to make decisions (produce translations) given incomplete knowledge. In such a case, a principled approach to solving that problem is to use the concepts of statistical decision theory to try to make optimal decisions given incomplete knowledge. This is the goal of statistical machine translation.

The use of statistical techniques in machine translation has led to dramatic improvements in the quality of research systems in recent years. For example, the statistical approaches of the VERBMOBIL evaluations (Wahlster 2000) or the U.S. National

* 1600 Amphitheatre Parkway, Mountain View, CA 94043. E-mail: och@google.com.
† Lehrstuhl für Informatik VI, Computer Science Department, RWTH Aachen–University of Technology, Ahornstr. 55, 52056 Aachen, Germany. E-mail: ney@cs.rwth-aachen.de.

Submission received: 19 November 2002; Revised submission received: 7 October 2003; Accepted for publication: 1 June 2004

Institute of Standards and Technology (NIST)/TIDES MT evaluations 2001 through 2003[1] obtain the best results. In addition, the field of statistical machine translation is rapidly progressing, and the quality of systems is getting better and better. An important factor in these improvements is definitely the availability of large amounts of data for training statistical models. Yet the modeling, training, and search methods have also improved since the field of statistical machine translation was pioneered by IBM in the late 1980s and early 1990s (Brown et al. 1990; Brown et al. 1993; Berger et al. 1994). This article focuses on an important improvement, namely, the use of (generalized) phrases instead of just single words as the core elements of the statistical translation model.

We describe in Section 2 the basics of our statistical translation model. We suggest the use of a log-linear model to incorporate the various knowledge sources into an overall translation system and to perform discriminative training of the free model parameters. This approach can be seen as a generalization of the originally suggested source–channel modeling framework for statistical machine translation.

In Section 3, we describe the statistical alignment models used to obtain a word alignment and techniques for learning phrase translations from word alignments. Here, the term *phrase* just refers to a consecutive sequence of words occurring in text and has to be distinguished from the use of the term in a linguistic sense. The learned bilingual phrases are not constrained by linguistic phrase boundaries. Compared to the word-based statistical translation models in Brown et al. (1993), this model is based on a (statistical) phrase lexicon instead of a single-word-based lexicon. Looking at the results of the recent machine translation evaluations, this approach seems currently to give the best results, and an increasing number of researchers are working on different methods for learning phrase translation lexica for machine translation purposes (Marcu and Wong 2002; Venugopal, Vogel, and Waibel 2003; Tillmann 2003; Koehn, Och, and Marcu 2003). Our approach to learning a phrase translation lexicon works in two stages: In the first stage, we compute an alignment between words, and in the second stage, we extract the aligned phrase pairs. In our machine translation system, we then use generalized versions of these phrases, called alignment templates, that also include the word alignment and use word classes instead of the words themselves.

In Section 4, we describe the various components of the statistical translation model. The backbone of the translation model is the alignment template feature function, which requires that a translation of a new sentence be composed of a set of alignment templates that covers the source sentence and the produced translation. Other feature functions score the well-formedness of the produced target language sentence (i.e., language model feature functions), the number of produced words, or the order of the alignment templates. Note that all components of our statistical machine translation model are purely data-driven and that there is no need for linguistically annotated corpora. This is an important advantage compared to syntax-based translation models (Yamada and Knight 2001; Gildea 2003; Charniak, Knight, and Yamada 2003) that require a parser for source or target language.

In Section 5, we describe in detail our search algorithm and discuss an efficient implementation. We use a dynamic-programming-based beam search algorithm that allows a trade-off between efficiency and quality. We also discuss the use of heuristic functions to reduce the number of search errors for a fixed beam size.

In Section 6, we describe various results obtained on different tasks. For the German–English VERBMOBIL task, we analyze the effect of various system compo-

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

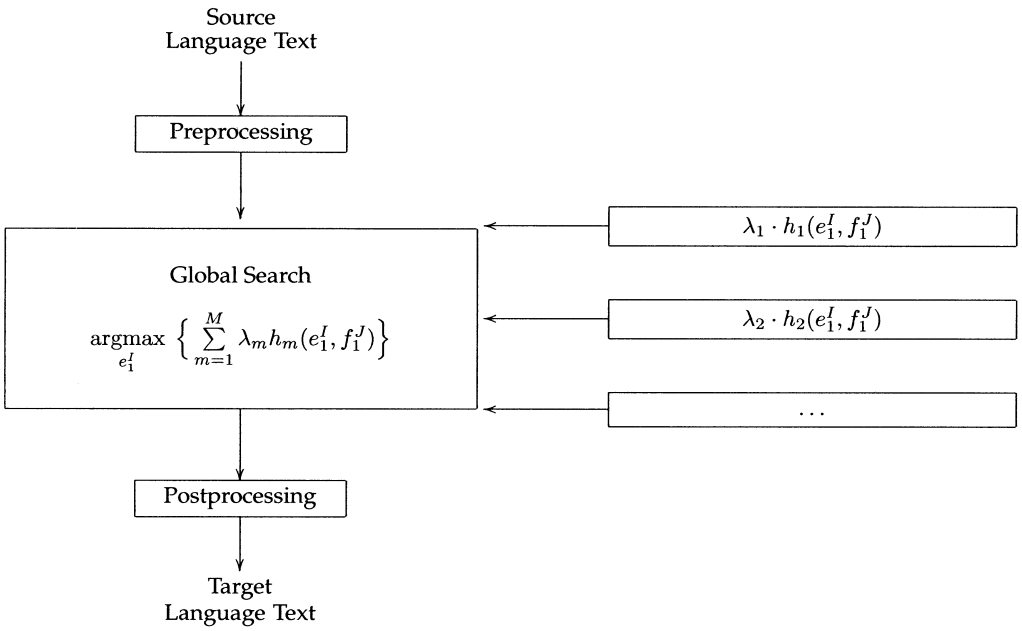

Figure 1
Architecture of the translation approach based on a log-linear modeling approach.

nents. On the French–English Canadian HANSARDS task, the alignment template system obtains significantly better results than a single-word-based translation model. In the Chinese–English 2002 NIST machine translation evaluation it yields results that are significantly better statistically than all competing research and commercial translation systems.

2. Log-Linear Models for Statistical Machine Translation

We are given a source (French) sentence $\mathbf{f} = f_1^J = f_1, \ldots, f_j, \ldots, f_J$, which is to be translated into a target (English) sentence $\mathbf{e} = e_1^I = e_1, \ldots, e_i, \ldots, e_I$. Among all possible target sentences, we will choose the sentence with the highest probability:[2]

$$\hat{e}_1^I = \underset{e_1^I}{\text{argmax}} \{Pr(e_1^I \mid f_1^J)\} \quad (1)$$

The argmax operation denotes the search problem, that is, the generation of the output sentence in the target language.

As an alternative to the often used source–channel approach (Brown et al. 1993), we directly model the posterior probability $Pr(e_1^I \mid f_1^J)$ (Och and Ney 2002). An especially well-founded framework for doing this is the maximum-entropy framework (Berger, Della Pietra, and Della Pietra 1996). In this framework, we have a set of M feature functions $h_m(e_1^I, f_1^J), m = 1, \ldots, M$. For each feature function, there exists a model

2 The notational convention employed in this article is as follows. We use the symbol $\Pr(\cdot)$ to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $p(\cdot)$.

parameter $\lambda_m, m = 1, \ldots, M$. The direct translation probability is given by

$$Pr(e_1^I \mid f_1^J) = p_{\lambda_1^M}(e_1^I \mid f_1^J) \quad (2)$$

$$= \frac{\exp[\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J)]}{\sum_{e'^I_1} \exp[\sum_{m=1}^M \lambda_m h_m(e'^I_1, f_1^J)]} \quad (3)$$

This approach has been suggested by Papineni, Roukos, and Ward (1997, 1998) for a natural language understanding task.

We obtain the following decision rule:

$$\hat{e}_1^I = \operatorname*{argmax}_{e_1^I} \left\{ Pr(e_1^I \mid f_1^J) \right\}$$

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

Hence, the time-consuming renormalization in equation (3) is not needed in search. The overall architecture of the log-linear modeling approach is summarized in Figure 1.

A standard criterion on a parallel training corpus consisting of S sentence pairs $\{(\mathbf{f}_s, \mathbf{e}_s)\colon\ s = 1, \ldots, S\}$ for log-linear models is the maximum class posterior probability criterion, which can be derived from the maximum-entropy principle:

$$\hat{\lambda}_1^M = \operatorname*{argmax}_{\lambda_1^M} \left\{ \sum_{s=1}^S \log p_{\lambda_1^M}(\mathbf{e}_s \mid \mathbf{f}_s) \right\} \quad (4)$$

This corresponds to maximizing the equivocation or maximizing the likelihood of the direct-translation model. This direct optimization of the posterior probability in Bayes' decision rule is referred to as discriminative training (Ney 1995) because we directly take into account the overlap in the probability distributions. The optimization problem under this criterion has very nice properties: There is one unique global optimum, and there are algorithms (e.g. gradient descent) that are guaranteed to converge to the global optimum. Yet the ultimate goal is to obtain good translation quality on unseen test data. An alternative training criterion therefore directly optimizes translation quality as measured by an automatic evaluation criterion (Och 2003).

Typically, the translation probability $Pr(e_1^I \mid f_1^J)$ is decomposed via additional hidden variables. To include these dependencies in our log-linear model, we extend the feature functions to include the dependence on the additional hidden variable. Using for example the alignment a_1^J as hidden variable, we obtain M feature functions of the form $h_m(e_1^I, f_1^J, a_1^J), m = 1, \ldots, M$ and the following model:

$$Pr(e_1^I, a_1^J \mid f_1^J) = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J, a_1^J)\right)}{\sum_{e'^I_1, a'^J_1} \exp\left(\sum_{m=1}^M \lambda_m h_m(e'^I_1, f_1^J, a'^J_1)\right)}$$

Obviously, we can perform the same step for translation models with an even richer set of hidden variables than only the alignment a_1^J.

3. Learning Translation Lexica

In this section, we describe methods for learning the single-word and phrase-based translation lexica that are the basis of the machine translation system described in

Section 4. First, we introduce the basic concepts of statistical alignment models, which are used to learn word alignment. Then, we describe how these alignments can be used to learn bilingual phrasal translations.

3.1 Statistical Alignment Models

In (statistical) alignment models $Pr(f_1^J, a_1^J \mid e_1^I)$, a "hidden" alignment $\mathbf{a} = a_1^J$ is introduced that describes a mapping from a source position j to a target position a_j. The relationship between the translation model and the alignment model is given by

$$Pr(f_1^J \mid e_1^I) = \sum_{a_1^J} Pr(f_1^J, a_1^J \mid e_1^I) \tag{5}$$

The alignment a_1^J may contain alignments $a_j = 0$ with the "empty" word e_0 to account for source words that are not aligned with any target word.

In general, the statistical model depends on a set of unknown parameters θ that is learned from training data. To express the dependence of the model on the parameter set, we use the following notation:

$$Pr(f_1^J, a_1^J \mid e_1^I) = p_\theta(f_1^J, a_1^J \mid e_1^I) \tag{6}$$

A detailed description of different specific statistical alignment models can be found in Brown et al. (1993) and Och and Ney (2003). Here, we use the hidden Markov model (HMM) alignment model (Vogel, Ney, and Tillmann 1996) and Model 4 of Brown et al. (1993) to compute the word alignment for the parallel training corpus.

To train the unknown parameters θ, we are given a parallel training corpus consisting of S sentence pairs $\{(\mathbf{f}_s, \mathbf{e}_s)\colon\ s = 1, \ldots, S\}$. For each sentence pair $(\mathbf{f}_s, \mathbf{e}_s)$, the alignment variable is denoted by $\mathbf{a} = a_1^J$. The unknown parameters θ are determined by maximizing the likelihood on the parallel training corpus:

$$\hat{\theta} \quad = \quad \underset{\theta}{\operatorname{argmax}} \left\{ \prod_{s=1}^{S} \left[\sum_{\mathbf{a}} p_\theta(\mathbf{f}_s, \mathbf{a} \mid \mathbf{e}_s) \right] \right\} \tag{7}$$

This optimization can be performed using the expectation maximization (EM) algorithm (Dempster, Laird, and Rubin 1977). For a given sentence pair there are a large number of alignments. The alignment $\hat{a}_1^J$ that has the highest probability (under a certain model) is also called the **Viterbi alignment** (of that model):

$$\hat{a}_1^J = \underset{a_1^J}{\operatorname{argmax}}\ p_{\hat{\theta}}(f_1^J, a_1^J \mid e_1^I) \tag{8}$$

A detailed comparison of the quality of these Viterbi alignments for various statistical alignment models compared to human-made word alignments can be found in Och and Ney (2003).

3.2 Symmetrization

The baseline alignment model does not allow a source word to be aligned with two or more target words. Therefore, lexical correspondences like the German compound word *Zahnarzttermin* for *dentist's appointment* cause problems because a single source word must be mapped onto two or more target words. Therefore, the resulting Viterbi alignment of the standard alignment models has a systematic loss in recall. Here, we

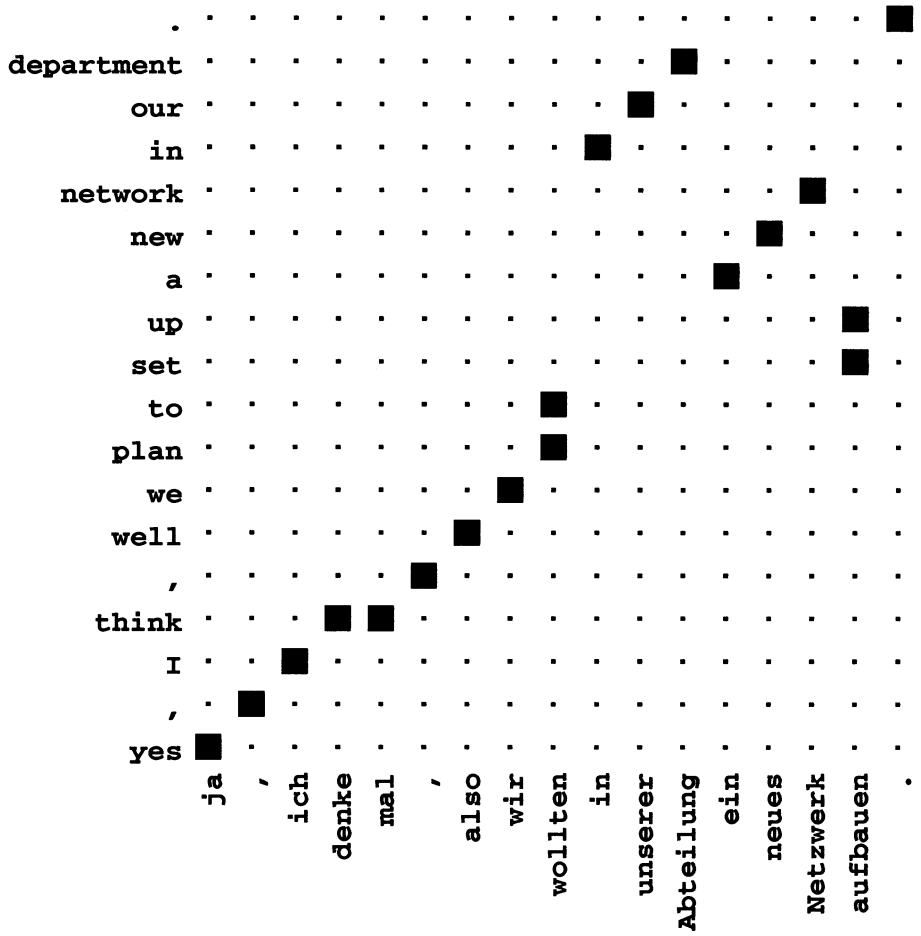

Figure 2
Example of a (symmetrized) word alignment (VERBMOBIL task).

describe various methods for performing a symmetrization of our directed statistical alignment models by applying a heuristic postprocessing step that combines the alignments in both translation directions (source to target, target to source). Figure 2 shows an example of a symmetrized alignment.

To solve this problem, we train in both translation directions. For each sentence pair, we compute two Viterbi alignments a_1^I and b_1^I. Let $A_1 = \{(a_j, j) \mid a_j > 0\}$ and $A_2 = \{(i, b_i) \mid b_i > 0\}$ denote the sets of alignments in the two Viterbi alignments. To increase the quality of the alignments, we can combine (symmetrize) A_1 and A_2 into one alignment matrix A using one of the following combination methods:

- Intersection: $A = A_1 \cap A_2$.
- Union: $A = A_1 \cup A_2$.
- Refined method: In a first step, the intersection $A = A_1 \cap A_2$ is determined. The elements of this intersection result from both Viterbi alignments and are therefore very reliable. Then, we extend the alignment A iteratively by adding alignments (i, j) occurring only in the

alignment A_1 or in the alignment A_2 if neither f_j nor e_i have an alignment in A, or if the following conditions both hold:

- The alignment (i,j) has a horizontal neighbor $(i-1,j)$, $(i+1,j)$ or a vertical neighbor $(i,j-1)$, $(i,j+1)$ that is already in A.
- The set $A \cup \{(i,j)\}$ does not contain alignments with both horizontal and vertical neighbors.

Obviously, the intersection yields an alignment consisting of only one-to-one alignments with a higher precision and a lower recall. The union yields a higher recall and a lower precision of the combined alignment. The refined alignment method is often able to improve precision and recall compared to the nonsymmetrized alignments. Whether a higher precision or a higher recall is preferred depends on the final application of the word alignment. For the purpose of statistical MT, it seems that a higher recall is more important. Therefore, we use the union or the refined combination method to obtain a symmetrized alignment matrix.

The resulting symmetrized alignments are then used to train single-word-based translation lexica $p(e \mid f)$ by computing relative frequencies using the count $N(e,f)$ of how many times e and f are aligned divided by the count $N(f)$ of how many times the word f occurs:

$$p(e \mid f) = \frac{N(e,f)}{N(f)}$$

3.3 Bilingual Contiguous Phrases

In this section, we present a method for learning relationships between whole phrases of m source language words and n target language words. This algorithm, which will be called `phrase-extract`, takes as input a general word alignment matrix (Section 3.2). The output is a set of bilingual phrases.

In the following, we describe the criterion that defines the set of phrases that **is consistent with** the word alignment matrix:

$$\begin{aligned} \mathcal{BP}(f_1^J, e_1^I, A) \quad = \quad \Big\{ \Big(f_j^{j+m}, e_i^{i+n}\Big) : \; & \forall (i',j') \in A : j \le j' \le j+m \leftrightarrow i \le i' \le i+n \\ & \wedge \exists (i',j') \in A : j \le j' \le j+m \wedge i \le i' \le i+n \Big\} \end{aligned} \tag{9}$$

Hence, the set of all bilingual phrases that are consistent with the alignment is constituted by all bilingual phrase pairs in which all words within the source language phrase are aligned only with the words of the target language phrase and the words of the target language phrase are aligned only with the words of the source language phrase. Note that we require that at least one word in the source language phrase be aligned with at least one word of the target language phrase. As a result there are no empty source or target language phrases that would correspond to the "empty word" of the word-based statistical alignment models.

These phrases can be computed straightforwardly by enumerating all possible phrases in one language and checking whether the aligned words in the other language are consecutive, with the possible exception of words that are not aligned at all. Figure 3 gives the algorithm `phrase-extract` that computes the phrases. The algorithm takes into account possibly unaligned words at the boundaries of the source or target language phrases. Table 1 shows the bilingual phrases containing between two and seven words that result from the application of this algorithm to the alignment of Figure 2.

Table 1
Examples of two- to seven-word bilingual phrases obtained by applying the algorithm `phrase-extract` to the alignment of Figure 2.

ja ,	yes ,
ja , ich	yes , I
ja , ich denke mal	yes , I think
ja , ich denke mal ,	yes , I think ,
ja , ich denke mal , also	yes , I think , well
, ich	, I
, ich denke mal	, I think
, ich denke mal ,	, I think ,
, ich denke mal , also	, I think , well
, ich denke mal , also wir	, I think , well we
ich denke mal	I think
ich denke mal ,	I think ,
ich denke mal , also	I think , well
ich denke mal , also wir	I think , well we
ich denke mal , also wir wollten	I think , well we plan to
denke mal ,	think ,
denke mal , also	think , well
denke mal , also wir	think , well we
denke mal , also wir wollten	think , well we plan to
, also	, well
, also wir	, well we
, also wir wollten	, well we plan to
also wir	well we
also wir wollten	well we plan to
wir wollten	we plan to
in unserer	in our
in unserer Abteilung	in our department
in unserer Abteilung ein neues Netzwerk	a new network in our department
in unserer Abteilung ein neues Netzwerk aufbauen	set up a new network in our department
unserer Abteilung	our department
ein neues	a new
ein neues Netzwerk	a new network
ein neues Netzwerk aufbauen	set up a new network
neues Netzwerk	new network

It should be emphasized that this constraint to consecutive phrases limits the expressive power. If a consecutive phrase in one language is translated into two or three nonconsecutive phrases in the other language, there is no corresponding bilingual phrase pair learned by this approach. In principle, this approach to learning phrases from a word-aligned corpus could be extended straightforwardly to handle nonconsecutive phrases in source and target language as well. Informal experiments have shown that allowing for nonconsecutive phrases significantly increases the number of extracted phrases and especially increases the percentage of wrong phrases. Therefore, we consider only consecutive phrases.

3.4 Alignment Templates

In the following, we add generalization capability to the bilingual phrase lexicon by replacing words with word classes and also by storing the alignment information for each phrase pair. These generalized and alignment-annotated phrase pairs are called **alignment templates**. Formally, an alignment template z is a triple $(F_1^{J'}, E_1^{I'}, \tilde{A})$

INPUT: e_1^I, f_1^J, A
$i_1 := 1$
WHILE $i_1 \leq I$
  $i_2 := i_1$
  WHILE $i_2 \leq I$
    $TP := \{j | \exists i : i_1 \leq i \leq i_2 \wedge A(i,j)\}$
    IF quasi-consecutive(TP)
    THEN
      $j_1 := \min(\text{TP})$
      $j_2 := \max(\text{TP})$
      $SP := \{i | \exists j : j_1 \leq j \leq j_2 \wedge A(i,j)\}$
      IF $SP \subseteq \{i_1, i_1 + 1, \ldots, i_2\}$
      THEN
        $\mathcal{BP} := \mathcal{BP} \cup \{(e_{i_1}^{i_2}, f_{j_1}^{j_2})\}$
        WHILE $j_1 > 0 \wedge \forall i : A(i, j_1) = 0$
          $j'' := j_2$
          WHILE $j'' \leq J \wedge \forall i : A(i, j'') = 0$
            $\mathcal{BP} := \mathcal{BP} \cup \{(e_{i_1}^{i_2}, f_{j_1}^{j''})\}$
            $j'' := j'' + 1$
          $j_1 := j_1 - 1$
OUTPUT: $\mathcal{BP}$

Figure 3
Algorithm `phrase-extract` for extracting phrases from a word-aligned sentence pair. Here quasi-consecutive(TP) is a predicate that tests whether the set of words TP is consecutive, with the possible exception of words that are not aligned.

that describes the alignment $\tilde{A}$ between a source class sequence $F_1^{J'}$ and a target class sequence $E_1^{I'}$. If each word corresponds to one class, an alignment template corresponds to a bilingual phrase together with an alignment within this phrase. Figure 4 shows examples of alignment templates.

The alignment $\tilde{A}$ is represented as a matrix with $J' \cdot (I' + 1)$ binary elements. A matrix element with value 1 means that the words at the corresponding positions are aligned, and the value 0 means that the words are not aligned. If a source word is not aligned with a target word, then it is aligned with the empty word e_0, which is at the imaginary position $i = 0$.

The classes used in $F_1^{J'}$ and $E_1^{I'}$ are automatically trained bilingual classes using the method described in Och (1999) and constitute a partition of the vocabulary of source and target language. In general, we are not limited to disjoint classes as long as each specific instance of a word is disambiguated, that is, uniquely belongs to a specific class. In the following, we use the class function C to map words to their classes. Hence, it would be possible to employ parts-of-speech or semantic categories instead of the automatically trained word classes used here.

The use of classes instead of the words themselves has the advantage of better generalization. For example, if there exist classes in source and target language that contain town names, it is possible that an alignment template learned using a specific town name can be generalized to other town names.

In the following, $\tilde{e}$ and $\tilde{f}$ denote target and source phrases, respectively. To train the probability of applying an alignment template $p(z = (F_1^{J'}, E_1^{I'}, \tilde{A}) \mid \tilde{f})$, we use an extended version of the algorithm `phrase-extract` from Section 3.3. All bilingual phrases that are consistent with the alignment are extracted together with the align-

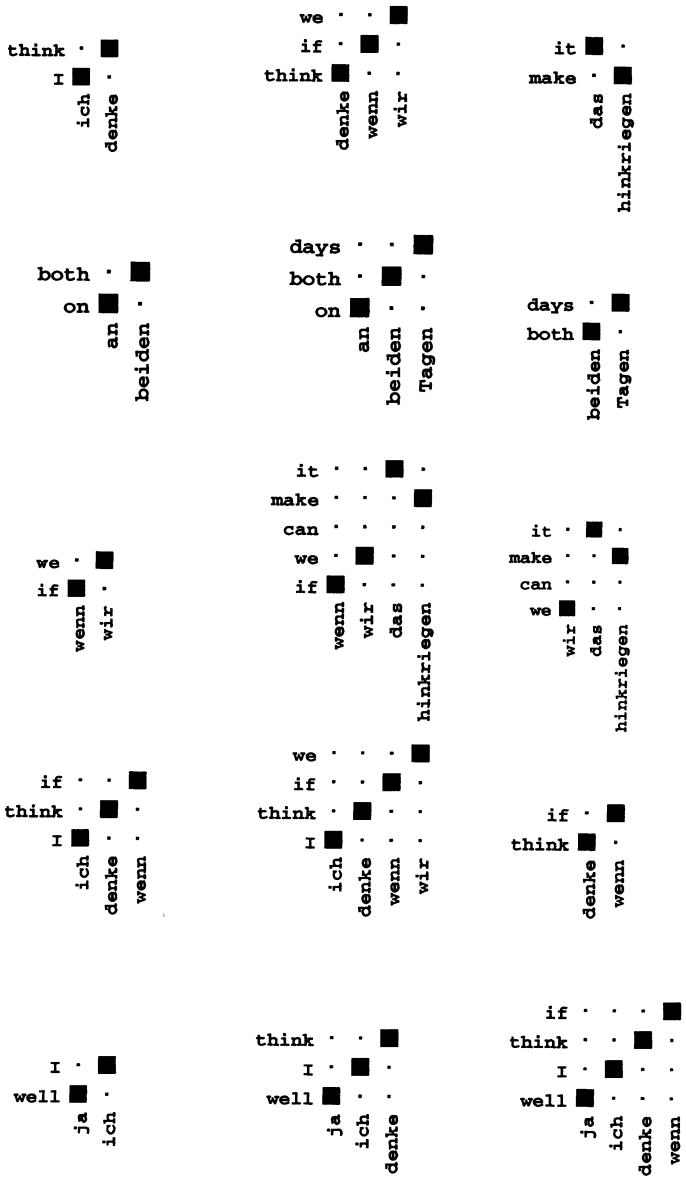

Figure 4
Examples of alignment templates obtained in training.

ment within this bilingual phrase. Thus, we obtain a count $N(z)$ of how often an alignment template occurred in the aligned training corpus. The probability of using an alignment template to translate a specific source language phrase $\tilde{f}$ is estimated by means of relative frequency:

$$p(z = (F_1^{J'}, E_1^{I'}, \tilde{A}) \mid \tilde{f}) = \frac{N(z) \cdot \delta(F_1^{J'}, C(\tilde{f}))}{N(C(\tilde{f}))} \tag{10}$$

To reduce the memory requirement of the alignment templates, we compute these probabilities only for phrases up to a certain maximal length in the source language.

Depending on the size of the corpus, the maximal length in the experiments is between four and seven words. In addition, we remove alignment templates that have a probability lower than a certain threshold. In the experiments, we use a threshold of 0.01.

It should be emphasized that this algorithm for computing aligned phrase pairs and their associated probabilities is very easy to implement. The joint translation model suggested by Marcu and Wong (2002) tries to learn phrases as part of a full EM algorithm, which leads to very large memory requirements and a rather complicated training algorithm. A comparison of the two approaches can be found in Koehn, Och, and Marcu (2003).

4. Translation Model

To describe our translation model based on the alignment templates described in the previous section in a formal way, we first decompose both the source sentence f_1^J and the target sentence e_1^I into a sequence of phrases ($k = 1, \ldots, K$):

$$f_1^J = \tilde{f}_1^K, \qquad \tilde{f}_k = f_{j_{k-1}+1}, \ldots, f_{j_k} \tag{11}$$

$$e_1^I = \tilde{e}_1^K, \qquad \tilde{e}_k = e_{i_{k-1}+1}, \ldots, e_{i_k} \tag{12}$$

Note that there are a large number of possible segmentations of a sentence pair into K phrase pairs. In the following, we will describe the model for a specific segmentation. Eventually, however, a model can be described in which the specific segmentation is not known when new text is translated. Hence, as part of the overall search process (Section 5), we also search for the optimal segmentation.

To allow possible reordering of phrases, we introduce an alignment on the phrase level π_1^K between the source phrases $\tilde{f}_1^K$ and the target phrases $\tilde{e}_1^K$. Hence, π_1^K is a permutation of the phrase positions $1, \ldots, K$ and indicates that the phrases $\tilde{e}_k$ and $\tilde{f}_{\pi_k}$ are translations of one another. We assume that for the translation between these phrases a specific alignment template z_k is used:

$$\tilde{e}_k \overset{z_k}{\longleftrightarrow} \tilde{f}_{\pi_k}$$

Hence, our model has the following hidden variables:

$$\pi_1^K, z_1^K$$

Figure 5 gives an example of the word alignment and phrase alignment of a German–English sentence pair.

We describe our model using a log-linear modeling approach. Hence, all knowledge sources are described as feature functions that include the given source language string f_1^J, the target language string e_1^I, and the above-stated hidden variables. Hence, we have the following functional form of all feature functions:

$$h(e_1^I, f_1^J, \pi_1^K, z_1^K)$$

Figure 6 gives an overview of the decisions made in the alignment template model. First, the source sentence words f_1^J are grouped into phrases $\tilde{f}_1^K$. For each phrase $\tilde{f}$ an alignment template z is chosen and the sequence of chosen alignment templates is reordered (according to π_1^K). Then, every phrase $\tilde{f}$ produces its translation $\tilde{e}$ (using the corresponding alignment template z). Finally, the sequence of phrases $\tilde{e}_1^K$ constitutes the sequence of words e_1^I.

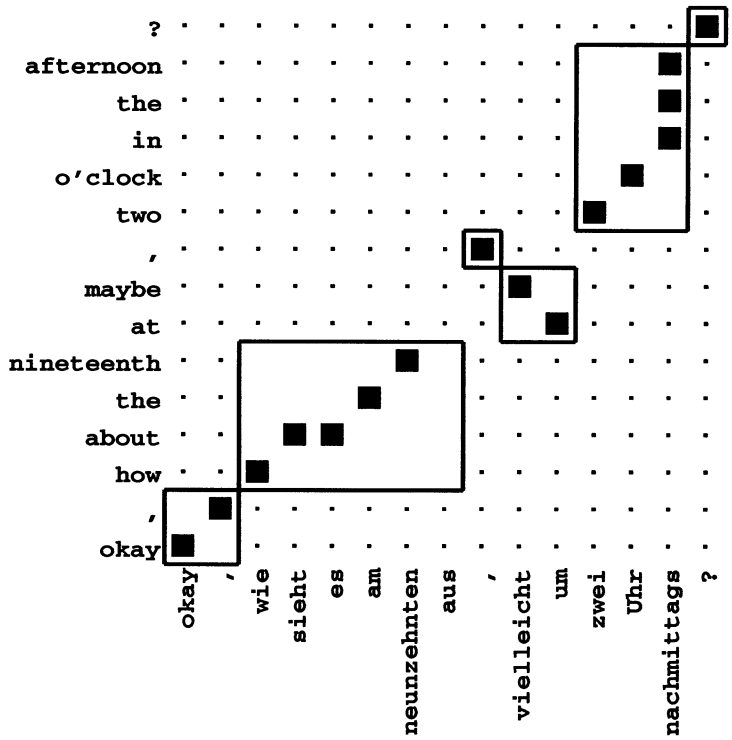

Figure 5
Example of segmentation of German sentence and its English translation into alignment templates.

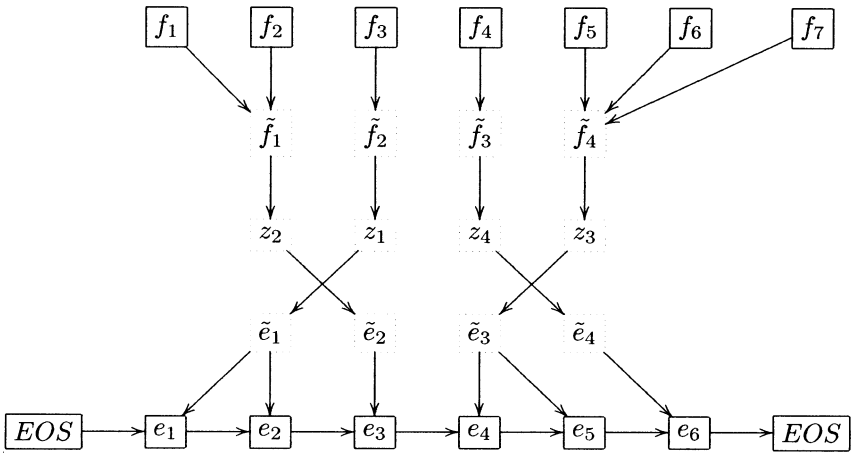

Figure 6
Dependencies in the alignment template model.

4.1 Feature Functions

4.1.1 Alignment Template Selection. To score the use of an alignment template, we use the probability $p(z \mid \tilde{f})$ defined in Section 3. We establish a corresponding feature function by multiplying the probability of all used alignment templates and taking the logarithm:

$$h_{\mathrm{AT}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \log \prod_{k=1}^{K} p(z_k \mid f_{j_{\pi_k-1}+1}^{j_{\pi_k}}) \tag{13}$$

Here, $j_{\pi_k-1}+1$ is the position of the first word of alignment template z_k in the source language sentence and j_{π_k} is the position of the last word of that alignment template.

Note that this feature function requires that a translation of a new sentence be composed of a set of alignment templates that covers both the source sentence and the produced translation. There is no notion of "empty phrase" that corresponds to the "empty word" in word-based statistical alignment models. The alignment on the phrase level is actually a permutation, and no insertions or deletions are allowed.

4.1.2 Word Selection. For scoring the use of target language words, we use a lexicon probability $p(e \mid f)$, which is estimated using relative frequencies as described in Section 3.2. The target word e depends on the aligned source words. If we denote the resulting word alignment matrix by $A := A_{\pi_1^K, z_1^K}$ and the predicted word class for word e_i by E_i, then the feature function h_{WRD} is defined as follows:

$$h_{\mathrm{WRD}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \log \prod_{i=1}^{I} p(e_i \mid \{f_j \mid (i,j) \in A\}, E_i) \tag{14}$$

For $p(e_i \mid \{f_j \mid (i,j) \in A\})$ we use a uniform mixture of a single-word model $p(e \mid f)$, which is constrained to predict only words that are in the predicted word class E_i:

$$p(e_i \mid \{f_j \mid (i,j) \in A\}, E_i) = \frac{\sum_{\{j \mid (i,j) \in A\}} p(e_i \mid f_j)}{|\{j \mid (i,j) \in A\}|} \cdot \delta(C(e_i), E_i)$$

A disadvantage of this model is that the word order is ignored in the translation model. The translations *the day after tomorrow* or *after the day tomorrow* for the German word *übermorgen* receive an identical probability. Yet the first one should obtain a significantly higher probability. Hence, we also include a dependence on the word positions in the lexicon model $p(e \mid f, i, j)$:

$$p(e_i \mid f_j, \sum_{i'=1}^{i-1}[(i', j) \in A], \sum_{j'=1}^{j-1}[(i, j') \in A]) \tag{15}$$

Here, $[(i', j) \in A]$ is 1 if $(i', j) \in A$ and 0 otherwise. As a result, the word e_i depends not only on the aligned French word f_j, but also on the number of preceding French words aligned with e_i and on the number of the preceding English words aligned with f_j. This model distinguishes the positions within a phrasal translation. The number of parameters of $p(e \mid f, i, j)$ is significantly higher than that of $p(e \mid f)$ alone. Hence, there is a data estimation problem especially for words that rarely occur. Therefore, we linearly interpolate the models $p(e \mid f)$ and $p(e \mid f, i, j)$.

4.1.3 Phrase Alignment. The phrase alignment feature simply takes into account that very often a monotone alignment is a correct alignment. Hence, the feature function h_{AL} measures the "amount of nonmonotonicity" by summing over the distance (in the

source language) of alignment templates that are consecutive in the target language:

$$h_{\mathrm{AL}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \sum_{k=1}^{K+1} |j_{\pi_k - 1} - j_{\pi_{k-1}}| \tag{16}$$

Here, j_{π_0} is defined to equal 0 and $j_{\pi_{K+1}-1}$ is defined to equal J. The above-stated sum includes $k = K + 1$ to include the distance from the end position of the last phrase to the end of sentence.

The sequence of $K = 6$ alignment templates in Figure 5 corresponds to the following sum of seven jump distances: $0 + 0 + 1 + 3 + 2 + 0 + 0 = 6$.

4.1.4 Language Model Features. As a default language model feature, we use a standard backing-off word-based trigram language model (Ney, Generet, and Wessel 1995):

$$h_{\mathrm{LM}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \log \prod_{i=1}^{I+1} p(e_i \mid e_{i-2}, e_{i-1}) \tag{17}$$

In addition, we use a 5-gram class-based language model:

$$h_{\mathrm{CLM}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \log \prod_{i=1}^{I+1} p(C(e_i) \mid C(e_{i-4}), \ldots, C(e_{i-1})) \tag{18}$$

The use of the language model feature in equation (18) helps take long-range dependencies better into account.

4.1.5 Word Penalty. To improve the scoring for different target sentence lengths, we also use as a feature the number of produced target language words (i.e., the length of the produced target language sentence):

$$h_{\mathrm{WP}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = I \tag{19}$$

Without this feature, we typically observe that the produced sentences tend to be too short.

4.1.6 Conventional Lexicon. We also use a feature that counts how many entries of a conventional lexicon co-occur in the given sentence pair. Therefore, the weight for the provided conventional dictionary can be learned:

$$h_{\mathrm{LEX}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \#\text{CO-OCCURRENCES}(\mathrm{LEX}, e_1^I, f_1^J) \tag{20}$$

The intuition is that the conventional dictionary LEX is more reliable than the automatically trained lexicon and therefore should get a larger weight.

4.1.7 Additional Features. A major advantage of the log-linear modeling approach used is that we can add numerous features that deal with specific problems of the baseline statistical MT system. Here, we will restrict ourselves to the described set of features. Yet we could use grammatical features that relate certain grammatical dependencies of source and target language. For example, using a function $k(\cdot)$ that counts how many arguments the main verb of a sentence has in the source or target sentence, we can define the following feature, which has a nonzero value if the verb in each of the two sentences has the same number of arguments:

$$h(f_1^J, e_1^I, \pi_1^K, z_1^K) = \delta(k(f_1^J), k(e_1^I)) \tag{21}$$

In the same way, we can introduce semantic features or pragmatic features such as the dialogue act classification.

4.2 Training

For the three different tasks on which we report results, we use two different training approaches. For the VERBMOBIL task, we train the model parameters λ_1^M according to the maximum class posterior probability criterion (equation (4)). For the French–English HANSARDS task and the Chinese–English NIST task, we simply tune the model parameters by coordinate descent on held-out data with respect to the automatic evaluation metric employed, using as a starting point the model parameters obtained on the VERBMOBIL task. Note that this tuning depends on the starting point of the model parameters and is not guaranteed to converge to the global optimum on the training data. As a result, this approach is limited to a very small number of model parameters. An efficient algorithm for performing this tuning for a larger number of model parameters can be found in Och (2003).

A standard approach to training the log-linear model parameters of the maximum class posterior probability criterion is the GIS (Generalized Iterative Scaling) algorithm (Darroch and Ratcliff 1972). To apply this algorithm, we have to solve various practical problems. The renormalization needed in equation (3) requires a sum over many possible sentences, for which we do not know of an efficient algorithm. Hence, we approximate this sum by extracting a large set of highly probable sentences as a sample from the space of all possible sentences (n-best approximation). The set of considered sentences is computed by means of an appropriately extended version of the search algorithm described in Section 5.

Using an n-best approximation, we might face the problem that the parameters trained with the GIS algorithm yield worse translation results even on the training corpus. This can happen because with the modified model scaling factors, the n-best list can change significantly and can include sentences that have not been taken into account in training. Using these sentences, the new model parameters might perform worse than the old model parameters. To avoid this problem, we proceed as follows. In a first step, we perform a search, compute an n-best list, and use this n-best list to train the model parameters. Second, we use the new model parameters in a new search and compute a new n-best list, which is combined with the existing n-best list. Third, using this extended n-best list, new model parameters are computed. This process is iterated until the resulting n-best list does not change. In this algorithm, convergence is guaranteed, as in the limit the n-best list will contain all possible translations. In practice, the algorithm converges after five to seven iterations. In our experiments this final n-best list contains about 500–1000 alternative translations.

We might have the problem that none of the given reference translations is part of the n-best list because the n-best list is too small or because the search algorithm performs pruning which in principle limits the possible translations that can be produced given a certain input sentence. To solve this problem, we define as reference translation for maximum-entropy training each sentence that has the minimal number of word errors with respect to any of the reference translations in the n-best list. More details of the training procedure can be found in Och and Ney (2002).

5. Search

In this section, we describe an efficient search architecture for the alignment template model.

5.1 General Concept

In general, the search problem for statistical MT even using only Model 1 of Brown et al. (1993) is NP-complete (Knight 1999). Therefore, we cannot expect to develop

efficient search algorithms that are guaranteed to solve the problem without search errors. Yet for practical applications it is acceptable to commit some search errors (Section 6.1.2). Hence, the art of developing a search algorithm lies in finding suitable approximations and heuristics that allow an efficient search without committing too many search errors.

In the development of the search algorithm described in this section, our main aim is that the search algorithm should be efficient. It should be possible to translate a sentence of reasonable length within a few seconds of computing time. We accept that the search algorithm sometimes results in search errors, as long as the impact on translation quality is minor. Yet it should be possible to reduce the number of search errors by increasing computing time. In the limit, it should be possible to search without search errors. The search algorithm should not impose any principal limitations. We also expect that the search algorithm be able to scale up to very long sentences with an acceptable computing time.

To meet these aims, it is necessary to have a mechanism that restricts the search effort. We accomplish such a restriction by searching in a breadth-first manner with pruning: beam search. In pruning, we constrain the set of considered translation candidates (the "beam") only to the promising ones. We compare in beam search those hypotheses that cover different parts of the input sentence. This makes the comparison of the probabilities problematic. Therefore, we integrate an admissible estimation of the remaining probabilities to arrive at a complete translation (Section 5.6)

Many of the other search approaches suggested in the literature do not meet the described aims:

- Neither optimal A* search (Och, Ueffing, and Ney 2001) nor optimal integer programming (Germann et al. 2001) for statistical MT allows efficient search for long sentences.
- Greedy search algorithms (Wang 1998; Germann et al. 2001) typically commit severe search errors (Germann et al. 2001).
- Other approaches to solving the search problem obtain polynomial time algorithms by assuming monotone alignments (Tillmann et al. 1997) or imposing a simplified recombination structure (Nießen et al. 1998). Others make simplifying assumptions about the search space (García-Varea, Casacuberta, and Ney 1998; García-Varea et al. 2001), as does the original IBM stack search decoder (Berger et al. 1994). All these simplifications ultimately make the search problem simpler but introduce fundamental search errors.

In the following, we describe our search algorithm based on the concept of beam search, which allows a trade-off between efficiency and quality by adjusting the size of the beam. The search algorithm can be easily adapted to other phrase-based translation models. For single-word-based search in MT, a similar algorithm has been described in Tillmann and Ney (2003).

5.2 Search Problem

Putting everything together and performing search in maximum approximation, we obtain the following decision rule:

$$\hat{e}_1^I = \underset{e_1^I, \pi_1^K, z_1^K}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m \cdot h_m(e_1^I, f_1^J, \pi_1^K, z_1^K) \right\} \tag{22}$$

Using the four feature functions AT, AL, WRD, and LM, we obtain the following decision rule:[3]

$$\hat{e}_1^I = \operatorname*{argmax}_{e_1^I, \pi_1^K, z_1^K} \Big\{ \tag{23}$$

$$\sum_{i=1}^{I} \left(\lambda_{\mathrm{LM}} \log p(e_i \mid e_{i-2}, e_{i-1}) + \lambda_{\mathrm{WRD}} \log p(e_i \mid \{f_j \mid (i,j) \in A\}, E_i) \right) \tag{24}$$

$$+ \sum_{k=1}^{K} \left(\lambda_{\mathrm{AT}} \log p(z_k \mid f_{j_{\pi_{k-1}}+1}^{j_{\pi_k}}) + \lambda_{\mathrm{AL}} \cdot |j_{\pi_k} - j_{\pi_{k-1}+1}| \right) \tag{25}$$

$$+ \lambda_{\mathrm{AL}} \cdot (J - j_{\pi_K}) + \lambda_{\mathrm{LM}} \log p(\mathrm{EOS} \mid e_{I-1}, e_I) \Big\} \tag{26}$$

Here, we have grouped the contributions of the various feature functions into those for each word (from LM and WRD, expression (24)), those for every alignment template (from AT and AL, expression (25)), and those for the end of sentence (expression (26)), which includes a term $\log p(\mathrm{EOS} \mid e_{I-1}, e_I)$ for the end-of-sentence language model probability.

To extend this decision rule for the word penalty (WP) feature function, we simply obtain an additional term λ_{WP} for each word. The class-based 5-gram language model (CLM) can be included like the trigram language model. Note that all these feature functions decompose nicely into contributions for each produced target language word or for each covered source language word. This makes it possible to develop an efficient dynamic programming search algorithm. Not all feature functions have this nice property: For the conventional lexicon feature function (LEX), we obtain an additional term in our decision rule which depends on the full sentence. Therefore, this feature function will not be integrated in the dynamic programming search but instead will be used to rerank the set of candidate translations produced by the search.

5.3 Structure of Search Space

We have to structure the search space in a suitable way to search efficiently. In our search algorithm, we generate search hypotheses that correspond to prefixes of target language sentences. Each hypothesis is the translation of a part of the source language sentence. A hypothesis is extended by appending one target word. The set of all hypotheses can be structured as a graph with a source node representing the sentence start, goal nodes representing complete translations, and intermediate nodes representing partial translations. There is a directed edge between hypotheses n_1 and n_2 if the hypothesis n_2 is obtained by appending one word to hypothesis n_1. Each edge has associated costs resulting from the contributions of all feature functions. Finally, our search problem can be reformulated as finding the optimal path through this graph.

In the first step, we determine the set of all source phrases in $\tilde{f}$ for which an applicable alignment template exists. Every possible application of an alignment template $z = (F_1^{J'}, E_1^{I'}, \tilde{A})$ to a subsequence $f_j^{j+J'-1}$ of the source sentence is called an **alignment template instantiation** $Z = (z, j)$. Hence, the set of all alignment template instantiations for the source sentence f_1^J is

$$\left\{ Z = (z,j) \mid z = (F_1^{J'}, E_1^{I'}, \tilde{A}) \wedge \exists j : p(z \mid f_j^{j+J'-1}) > 0 \right\} \tag{27}$$

3 Note that here some of the simplifying notation of Section 4 has been used.

If the source sentence contains words that have not been seen in the training data, we introduce a new alignment template that performs a one-to-one translation of each of these words by itself.

In the second step, we determine a set of probable target language words for each target word position in the alignment template instantiation. Only these words are then hypothesized in the search. We call this selection of highly probable words **observation pruning** (Tillmann and Ney 2000). As a criterion for a word e at position i in the alignment template instantiation, we use

$$\delta(E_i, C(e)) \cdot \sum_{j=0}^{J'} \frac{\tilde{A}(i,j)}{\sum_{i'} \tilde{A}(i',j)} \cdot p(e \mid f_j) \tag{28}$$

In our experiments, we hypothesize only the five best-scoring words.

A decision is a triple $d = (Z, e, l)$ consisting of an alignment template instantiation Z, the generated word e, and the index l of the generated word in Z. A hypothesis n corresponds to a valid sequence of decisions d_1^i. The possible decisions are as follows:

1. Start a new alignment template: $d_i = (Z_i, e_i, 1)$. In this case, the index $l = 1$. This decision can be made only if the previous decision d_{i-1} finished an alignment template and if the newly chosen alignment template instantiation does not overlap with any previously chosen alignment template instantiation. The resulting decision score corresponds to the contribution of the LM and the WRD features (expression (24)) for the produced word and the contribution of AL and AT features (expression (25)) for the started alignment template.
2. Extend an alignment template: $d_i = (Z_i, e_i, l)$. This decision can be made only if the previous decision uses the same alignment template instantiation and has as index $l - 1$: $d_{i-1} = (Z_i, e_{i-1}, l - 1)$. The resulting decision score corresponds to the contribution of the LM and the WRD features (expression (24)).
3. Finish the translation of a sentence: $d_i = (\mathrm{EOS}, \mathrm{EOS}, 0)$. In this case, the hypothesis is marked as a goal hypothesis. This decision is possible only if the previous decision d_{i-1} finished an alignment template and if the alignment template instantiations completely cover the input sentence. The resulting decision score corresponds to the contribution of expression (26).

Any valid and complete sequence of decisions d_1^{I+1} uniquely corresponds to a certain translation e_1^I, a segmentation into K phrases, a phrase alignment π_1^K, and a sequence of alignment template instantiations z_1^K. The sum of the decision scores is equal to the corresponding score described in expressions (24)–(26).

A straightforward representation of all hypotheses would be the prefix tree of all possible sequences of decisions. Obviously, there would be a large redundancy in this search space representation, because there are many search nodes that are indistinguishable in the sense that the subtrees following these search nodes are identical. We can recombine these identical search nodes; that is, we have to maintain only the most probable hypothesis (Bellman 1957).

In general, the criterion for recombining a set of nodes is that the hypotheses can be distinguished by neither language nor translation model. In performing recombination,

INPUT: implicitly defined search space (functions `Recombine`, `Extend`)

$H = \{\text{initial-hypothesis}\}$

WHILE $H \neq \emptyset$

- $H_{ext} := \emptyset$
- FOR $n \in H$
  - IF hypothesis n is final
  - THEN $H_{fin} := H_{fin} \cup \{n\}$
  - ELSE $H_{ext} := H_{ext} \cup \texttt{Extend}(n)$
- $H := \texttt{Recombine}(H_{ext})$
- $\hat{Q} = \max_{n \in H} Q(n)$
- $H := \{n \in H : Q(n) > \log(t_p) + \hat{Q}\}$
- $H := \text{HistogramPruning}(H, N_p)$

$\hat{n} = \underset{n \in H_{fin}}{\text{argmax}}\ Q(n)$

OUTPUT: $\hat{n}$

Figure 7
Algorithm for breadth-first search with pruning.

we obtain a search graph instead of a search tree. The exact criterion for performing recombination for the alignment templates is described in Section 5.5.

5.4 Search Algorithm

Theoretically, we could use any graph search algorithm to search the optimal path in the search space. We use a breadth-first search algorithm with pruning. This approach offers very good possibilities for adjusting the trade-off between quality and efficiency. In pruning, we always compare hypotheses that have produced the same number of target words.

Figure 7 shows a structogram of the algorithm. As the search space increases exponentially, it is not possible to explicitly represent it. Therefore, we represent the search space implicitly, using the functions `Extend` and `Recombine`. The function `Extend` produces new hypotheses extending the current hypothesis by one word. Some hypotheses might be identical or indistinguishable by the language and translation models. These are recombined by the function `Recombine`. We expand the search space such that only hypotheses with the same number of target language words are recombined.

In the pruning step, we use two different types of pruning. First, we perform pruning relative to the score $\hat{Q}$ of the current best hypothesis. We ignore all hypotheses that have a probability lower than $\log(t_p) + \hat{Q}$, where t_p is an adjustable pruning parameter. This type of pruning can be performed when the hypothesis extensions are computed. Second, in histogram pruning (Steinbiss, Tran, and Ney 1994), we maintain only the best N_p hypotheses. The two pruning parameters t_p and N_p have to be optimized with respect to the trade-off between efficiency and quality.

5.5 Implementation

In this section, we describe various issues involved in performing an efficient implementation of a search algorithm for the alignment template approach.

A very important design decision in the implementation is the representation of a hypothesis. Theoretically, it would be possible to represent search hypotheses only by the associated decision and a back-pointer to the previous hypothesis. Yet this would be a very inefficient representation for the implementation of the operations

that have to be performed in the search. The hypothesis representation should contain all information required to perform efficiently the computations needed in the search but should contain no more information than that, to keep the memory consumption small.

In search, we produce hypotheses n, each of which contains the following information:

1. e: the final target word produced
2. h: the state of the language model (to predict the following word)
3. $\mathbf{c} = c_1^J$: the coverage vector representing the already covered positions of the source sentence ($c_j = 1$ means the position j is covered, $c_j = 0$ means the position j is not covered)
4. Z: a reference to the alignment template instantiation that produced the final target word
5. l: the position of the final target word in the alignment template instantiation
6. $Q(n)$: the accumulated score of all previous decisions
7. n': a reference to the previous hypothesis

Using this representation, we can perform the following operations very efficiently:

- Determining whether a specific alignment template instantiation can be used to extend a hypothesis. To do this, we check whether the positions of the alignment template instantiation are still free in the hypothesis coverage vector.
- Checking whether a hypothesis is final. To do this, we determine whether the coverage vector contains no uncovered position. Using a bit vector as representation, the operation to check whether a hypothesis is final can be implemented very efficiently.
- Checking whether two hypotheses can be recombined. The criterion for recombining two hypotheses $n_1 = (e_1, h_1, \mathbf{c}_1, Z_1, l_1)$ and $n_2 = (e_2, h_2, \mathbf{c}_2, Z_2, l_2)$ is

$$
\begin{array}{ll}
h_1 = h_2 \wedge & \text{identical language model state} \\
\mathbf{c}_1 = \mathbf{c}_2 \wedge & \text{identical coverage vector} \\
\left(\quad (Z_1 = Z_2 \wedge l_1 = l_2) \vee \right. & \text{alignment template instantiation is identical} \\
\left. \quad (J(Z_1) = l_1 \wedge J(Z_2) = l_2) \quad \right) & \text{alignment template instantiation finished}
\end{array}
$$

We compare in beam search those hypotheses that cover different parts of the input sentence. This makes the comparison of the probabilities problematic. Therefore, we integrate an admissible estimation of the remaining probabilities to arrive at a complete translation. Details of the heuristic function for the alignment templates are provided in the next section.

5.6 Heuristic Function

To improve the comparability of search hypotheses, we introduce heuristic functions. A heuristic function estimates the probabilities of reaching the goal node from a certain

search node. An **admissible** heuristic function is always an optimistic estimate; that is, for each search node, the product of edge probabilities of reaching a goal node is always equal to or smaller than the estimated probability. For an A*-based search algorithm, a good heuristic function is crucial to being able to translate long sentences. For a beam search algorithm, the heuristic function has a different motivation. It is used to improve the scoring of search hypotheses. The goal is to make the probabilities of all hypotheses more comparable, in order to minimize the chance that the hypothesis leading to the optimal translation is pruned away.

Heuristic functions for search in statistical MT have been used in Wang and Waibel (1997) and Och, Ueffing, and Ney (2001). Wang and Waibel (1997) have described a simple heuristic function for Model 2 of Brown et al. (1993) that was not admissible. Och, Ueffing, and Ney (2001) have described an admissible heuristic function for Model 4 of Brown et al. (1993) and an almost-admissible heuristic function that is empirically obtained.

We have to keep in mind that a heuristic function is helpful only if the overhead introduced in computing the heuristic function is more than compensated for by the gain obtained through a better pruning of search hypotheses. The heuristic functions described in the following are designed such that their computation can be performed efficiently.

The basic idea for developing a heuristic function for an alignment model is that all source sentence positions that have not been covered so far still have to be translated to complete the sentence. If we have an estimation $r^X(j)$ of the optimal score for translating position j, then the value of the heuristic function $R^X(n)$ for a node n can be inferred by summing over the contribution for every position j that is not in the coverage vector $c(n)$ (here X denotes different possibilities to choose the heuristic function):

$$R^X(n) = \sum_{j \notin c(n)} r^X(j) \tag{29}$$

The situation in the case of the alignment template approach is more complicated, as not every word is translated alone, but typically the words are translated in context. Therefore, the basic quantity for the heuristic function in the case of the alignment template approach is a function $r(Z)$ that assigns to every alignment template instantiation Z a maximal probability. Using $r(Z)$, we can induce a position-dependent heuristic function $r(j)$:

$$r(j) := \max_{Z: j(Z) \leq j \leq j(Z)+J(Z)-1} r(Z)/J(Z) \tag{30}$$

Here, $J(Z)$ denotes the number of source language words produced by the alignment template instantiation Z and $j(Z)$ denotes the position of the first source language word. It can be easily shown that if $r(Z)$ is admissible, then $r(j)$ is also admissible. We have to show that for all nonoverlapping sequences Z_1^K the following holds:

$$\sum_{k=1}^{K} r(Z_k) \leq \sum_{j \in c(Z_1^K)} r(j) \tag{31}$$

Here, $c(Z_1^K)$ denotes the set of all positions covered by the sequence of alignment templates Z_1^K. This can be shown easily:

$$\sum_{k=1}^{K} r(Z_k) \quad = \quad \sum_{k=1}^{K} \sum_{j=1}^{J(Z_k)} r(Z_k)/J(Z_k) \tag{32}$$

INPUT: coverage vector c_1^J, previously covered position j
$\text{ff} = \min(\{j' \mid c_{j'} = 0\})$
$\text{mj} = \lvert j - \text{ff} \rvert$
WHILE $\text{ff} \neq (J+1)$
$\quad \text{fo} := \min(\{j' \mid j' > \text{ff} \wedge c_{j'} = 1\})$
$\quad \text{ff} := \min(\{j' \mid j' > \text{fo} \wedge c_{j'} = 0 \vee j' = J+1\})$
$\quad \text{mj} := \text{mj} + \lvert \text{ff} - \text{fo} \rvert$
OUTPUT: mj

Figure 8
Algorithm `min-jumps` to compute the minimum number of needed jumps $D(c_1^J, j)$ to complete the translation.

$$= \sum_{j \in c(Z_1^K)} r(Z_{k(j)})/J(Z_{k(j)}) \quad (33)$$

$$\leq \sum_{j \in c(Z_1^K)} \max_{Z: j(Z) \leq j \leq j(Z)+J(Z)-1} r(Z)/J(Z) \quad (34)$$

Here, $k(j)$ denotes the phrase index k that includes the target language word position j. In the following, we develop various heuristic functions $r(Z)$ of increasing complexity. The simplest realization of a heuristic function $r(Z)$ takes into account only the prior probability of an alignment template instantiation:

$$R^{\text{AT}}(Z = (z, j)) = \lambda_{\text{AT}} \cdot \log p(z \mid f_{j,j+J(z)-1}) \quad (35)$$

The lexicon model can be integrated as follows:

$$R^{\text{WRD}}(Z) = \lambda_{\text{WRD}} \cdot \sum_{j'=j(Z)}^{j(Z)+J(Z)-1} \max_e \log p(e \mid f_{j'}) \quad (36)$$

The language model can be incorporated by considering that for each target word there exists an optimal language model probability:

$$p^{\text{L}}(e) = \max_{e', e''} p(e \mid e', e'') \quad (37)$$

Here, we assume a trigram language model. In general, it is necessary to maximize over all possible different language model histories. We can also combine the language model and the lexicon model into one heuristic function:

$$R^{\text{WRD+LM}}(Z) = \sum_{j'=j(Z)}^{j(Z)+J(Z)-1} \max_e \lambda_{\text{WRD}} \log(p(e \mid f_{j'})) + \lambda_{\text{LM}} \log(p^{\text{L}}(e)) \quad (38)$$

To include the phrase alignment probability in the heuristic function, we compute the minimum sum of all jump widths that is needed to complete the translation. This sum can be computed efficiently using the algorithm shown in Figure 8. Then, an admissible heuristic function for the jump width is obtained by

$$R^{\text{AL}}(\mathbf{c}, j) = \lambda_{\text{AL}} \cdot D(\mathbf{c}, j) \quad (39)$$

Table 2
Statistics for VERBMOBIL task: training corpus (Train), conventional dictionary (Lex), development corpus (Dev), test corpus (Test) (Words*: words without punctuation marks).

		No Preprocessing		With Preprocessing	
		German	English	German	English
Train	Sentences		58,073		
	Words	519,523	549,921	522,933	548,874
	Words*	418,974	453,612	420,919	450,297
	Singletons	3,453	1,698	3,570	1,763
	Vocabulary	7,940	4,673	8,102	4,780
Lex	Entries		12,779		
	Extended vocabulary	11,501	6,867	11,904	7,089
Dev	Sentences		276		
	Words	3,159	3,438	3,172	3,445
	Trigram perplexity	—	28.1	—	26.3
Test	Sentences		251		
	Words	2,628	2,871	2,640	2,862
	Trigram perplexity	—	30.5	—	29.9

Combining all the heuristic functions for the various models, we obtain as final heuristic function for a search hypothesis n

$$R(n) = R^{\mathrm{AL}}(c(n), j(n)) + \sum_{j \notin c(n)} \left(R^{\mathrm{AT}}(j) + R^{\mathrm{WRD+LM}}(j) \right) \tag{40}$$

6. Results

6.1 Results on the VERBMOBIL Task

We present results on the VERBMOBIL task, which is a speech translation task in the domain of appointment scheduling, travel planning, and hotel reservation (Wahlster 2000). Table 2 shows the corpus statistics for this task. We use a training corpus, which is used to train the alignment template model and the language models, a development corpus, which is used to estimate the model scaling factors, and a test corpus. On average, 3.32 reference translations for the development corpus and 5.14 reference translations for the test corpus are used.

A standard vocabulary had been defined for the various speech recognizers used in VERBMOBIL. However, not all words of this vocabulary were observed in the training corpus. Therefore, the translation vocabulary was extended semiautomatically by adding about 13,000 German–English entries from an online bilingual lexicon available on the Web. The resulting lexicon contained not only word-word entries, but also multi-word translations, especially for the large number of German compound words. To counteract the sparseness of the training data, a couple of straightforward rule-based preprocessing steps were applied *before* any other type of processing:

- normalization of
  - numbers
  - time and date phrases
  - spelling (e.g., *don't* → *do not*)
- splitting of German compound words.

So far, in machine translation research there is no generally accepted criterion for the evaluation of experimental results. Therefore, we use various criteria. In the following experiments, we use:

- WER (word error rate)/mWER (multireference 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. In the case of the multireference word error rate for each test sentence, not just a single reference translation is used, as for the WER, but a whole set of reference translations. For each translation hypothesis, the edit distance to the most similar sentence is calculated (Nießen et al. 2000).
- PER (position-independent WER): A shortcoming of the WER is the fact that it requires a perfect word order. An acceptable sentence can have a word order that is different from that of the target sentence, so the WER measure alone could be misleading. To overcome this problem, we introduce as an additional measure the position-independent word error rate. This measure compares the words in the two sentences, ignoring the word order.
- BLEU (bilingual evalutation understudy) score: This score measures the precision of unigrams, bigrams, trigrams, and 4-grams with respect to a whole set of reference translations, with a penalty for too-short sentences (Papineni et al. 2001). Unlike all other evaluation criteria used here, BLEU measures accuracy, that is, the opposite of error rate. Hence, the larger BLEU scores, the better.

In the following, we analyze the effect of various system components: alignment template length, search pruning, and language model n-gram size. A systematic evaluation of the alignment template system comparing it with other translation approaches (e.g., rule-based) has been performed in the VERBMOBIL project and is described in Tessiore and von Hahn (2000). There, the alignment-template-based system achieved a significantly larger number of "approximately correct" translations than the competing translation systems (Ney, Och, and Vogel 2001).

6.1.1 Effect of Alignment Template Length. Table 3 shows the effect of constraining the maximum length of the alignment templates in the source language. Typically, it is necessary to restrict the alignment template length to keep memory requirements low. We see that using alignment templates with only one or two words in the source languages results in very bad translation quality. Yet using alignment templates with lengths as small as three words yields optimal results.

6.1.2 Effect of Pruning and Heuristic Function. In the following, we analyze the effect of beam search pruning and of the heuristic function. We use the following criteria:

- **Number of search errors:** A search error occurs when the search algorithm misses the most probable translation and produces a translation which is less probable. As we typically cannot efficiently compute the probability of the optimal translation, we cannot efficiently compute the number of search errors. Yet we can compute a lower bound on the number of search errors by comparing the translation

Table 3
Effect of alignment template length on translation quality.

AT length	PER [%]	mWER [%]	BLEU [%]
1	29.8	39.9	44.6
2	27.0	33.0	53.6
3	26.5	30.7	56.1
4	26.9	31.4	55.7
5	26.8	31.4	55.7
6	26.5	30.9	56.0
7	26.5	30.9	56.1

Table 4
Effect of pruning parameter t_p and heuristic function on search efficiency for direct-translation model ($N_p = 50{,}000$).

	no heuristic function		AT+WRD		+LM		+AL	
	time	search	time	search	time	search	time	search
t_p	[s]	errors	[s]	errors	[s]	errors	[s]	errors
10^{-2}	0.0	194	0.0	174	0.0	150	0.0	91
10^{-4}	0.2	97	0.2	57	0.3	40	0.2	13
10^{-6}	2.0	61	2.8	21	4.1	11	1.8	3
10^{-8}	11.9	41	15.0	7	19.9	5	9.5	1
10^{-10}	45.6	38	50.9	6	65.2	3	32.0	1
10^{-12}	114.6	34	119.2	5	146.2	2	75.2	0

found under specific pruning thresholds with the best translation that we have found using very conservative pruning thresholds.

- **Average translation time per sentence:** Pruning is used to adjust the trade-off between efficiency and quality. Hence, we present the average time needed to translate one sentence of the test corpus.
- **Translation quality (mWER, BLEU):** Typically, a sentence can have many different correct translations. Therefore, a search error does not necessarily result in poorer translation quality. It is even possible that a search error can improve translation quality. Hence, we analyze the effect of search on translation quality, using the automatic evaluation criteria mWER and BLEU.

Tables 4 and 5 show the effect of the pruning parameter t_p with the histogram pruning parameter $N_p = 50{,}000$. Tables 6 and 7 show the effect of the pruning parameter N_p with the pruning parameter $t_p = 10^{-12}$. In all four tables, we provide the results for using no heuristic functions and three variants of an increasingly informative heuristic function. The first is an estimate of the alignment template and the lexicon probability (AT+WRD), the second adds an estimate of the language model (+LM) probability, and the third also adds the alignment probability (+AL). These heuristic functions are described in Section 5.6.

Without a heuristic function, even more than a hundred seconds per sentence cannot guarantee search-error-free translation. We draw the conclusion that a good heuristic function is very important to obtaining an efficient search algorithm.

Table 5
Effect of pruning parameter t_p and heuristic function on error rate for direct-translation model ($N_p = 50{,}000$).

	error rates [%]							
	no heuristic function		AT+WRD		+LM		+AL	
t_p	mWER	BLEU	mWER	BLEU	mWER	BLEU	mWER	BLEU
10^{-2}	48.9	46.8	44.3	49.4	40.9	51.3	33.6	53.4
10^{-4}	39.8	50.9	35.0	53.8	32.3	55.0	30.7	55.9
10^{-6}	37.1	51.3	31.8	55.0	30.9	55.6	30.8	56.0
10^{-8}	35.7	53.0	31.4	55.7	31.2	55.7	30.9	56.0
10^{-10}	36.1	52.9	31.3	55.8	31.0	55.9	30.8	56.0
10^{-12}	35.7	52.9	31.2	55.9	31.0	55.9	30.8	56.0

Table 6
Effect of pruning parameter N_p and heuristic function on search efficiency for direct-translation model ($t_p = 10^{-12}$).

	no heuristic function		AT+WRD		+LM		+AL	
	time	search	time	search	time	search	time	search
N_p	[s]	errors	[s]	errors	[s]	errors	[s]	errors
1	0.0	237	0.0	238	0.0	238	0.0	232
10	0.0	169	0.0	154	0.0	148	0.0	98
100	0.3	101	0.3	69	0.3	60	0.2	21
1,000	2.2	65	2.3	33	2.4	27	2.0	5
10,000	18.3	40	18.3	10	21.1	5	14.3	1
50,000	114.6	34	119.2	5	146.2	2	75.2	0

Table 7
Effect of pruning parameter N_p and heuristic function on error rate for direct-translation model ($t_p = 10^{-12}$).

	error rates [%]							
	no heuristic function		AT+WRD		+LM		+AL	
N_p	mWER	BLEU	mWER	BLEU	mWER	BLEU	mWER	BLEU
1	64.4	29.7	61.9	31.7	59.8	32.4	49.4	38.2
10	46.6	46.9	43.0	49.2	42.0	49.1	34.6	52.3
100	41.0	49.8	36.7	52.6	34.8	53.8	31.3	55.6
1,000	37.8	51.5	33.0	54.5	32.3	55.3	30.6	56.0
10,000	35.5	53.1	31.4	55.6	30.9	55.6	30.8	56.0
50,000	35.7	52.9	31.2	55.9	31.0	55.9	30.8	56.0

Table 8
Effect of the length of the language model history (Unigram/Bigram/Trigram: word-based; CLM: class-based 5-gram).

Language model type	PP	PER [%]	mWER [%]	BLEU [%]
Zerogram	4781.0	38.1	45.9	29.0
Unigram	203.1	30.2	40.9	37.7
Bigram	38.3	26.9	32.9	53.0
Trigram	29.9	26.8	31.8	55.2
Trigram + CLM	—	26.5	30.9	56.1

In addition, the search errors have a more severe effect on the error rates if we do not use a heuristic function. If we compare the error rates in Table 7, which correspond to about 55 search errors in Table 6, we obtain an mWER of 36.7% (53 search errors) using no heuristic function and an mWER of 32.6% (57 search errors) using the combined heuristic function. The reason is that without a heuristic function, often the "easy" part of the input sentence is translated first. This yields severe reordering errors.

6.1.3 Effect of the Length of the Language Model History. In this work, we use only n-gram-based language models. Ideally, we would like to take into account long-range dependencies. Yet long n-grams are seen rarely and are therefore rarely used on unseen data. Therefore, we expect that extending the history length will at some point not improve further translation quality.

Table 8 shows the effect of the length of the language model history on translation quality. We see that the language model perplexity improves from 4,781 for a unigram model to 29.9 for a trigram model. The corresponding translation quality improves from an mWER of 45.9% to an mWER of 31.8%. The largest effect seems to come from taking into account the bigram dependence, which achieves an mWER of 32.9%. If we perform log-linear interpolation of a trigram model with a class-based 5-gram model, we observe an additional small improvement in translation quality to an mWER of 30.9%.

6.2 Results on the HANSARDS task

The HANSARDS task involves the proceedings of the Canadian parliament, which are kept by law in both French and English. About three 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 of up to 30 words. Table 9 shows the training and test corpus statistics.

The results for French to English and for English to French are shown in Table 10. Because of memory limitations, the maximum alignment template length has been restricted to four words. We compare here against the single-word-based search for Model 4 described in Tillmann (2001). We see that the alignment template approach obtains significantly better results than the single-word-based search.

6.3 Results on Chinese–English

Various statistical, example-based, and rule-based MT systems for a Chinese–English news domain were evaluated in the NIST 2002 MT evaluation.[4] Using the alignment

4 Evaluation home page: http://www.nist.gov/speech/tests/mt/mt2001/index.htm.

Table 9
Corpus statistics for HANSARDS task (Words*: words without punctuation marks).

		French	English
Training	Sentences	1,470,473	
	Words	24,338,195	22,163,092
	Words*	22,175,069	20,063,378
	Vocabulary	100,269	78,332
	Singletons	40,199	31,319
Test	Sentences	5,432	
	Words	97,646	88,773
	Trigram perplexity	—	179.8

Table 10
Translation results on the HANSARDS task.

	French→English		English→French	
Translation approach	WER [%]	PER [%]	WER [%]	PER [%]
Alignment templates	61.5	49.2	60.9	47.9
Single-word-based: monotone search	65.5	53.0	66.6	56.3
Single-word-based: reordering search	64.9	51.4	66.0	54.4

Table 11
Corpus statistics for Chinese–English corpora—large data track (Words*: words without punctuation marks).

		No preprocessing		With preprocessing	
		Chinese	English	Chinese	English
Train	Sentences	1,645,631			
	Unique sentences	1,289,890			
	Words	31,175,023	33,044,374	30,849,149	32,511,418
	Words*	27,091,283	29,212,384	26,828,721	28,806,735
	Singletons	15,324	24,933	5,336	26,344
	Vocabulary	67,103	92,488	45,111	85,116
Lex	Entries	80,977			
	Extended vocabulary	76,182	100,704	54,190	93,350
Dev	Sentences	993			
	Words	26,361	32,267	25,852	31,607
	Trigram perplexity	—	237,154	—	171,922
Test	Sentences	878			
	Words	24,540	—	24,144	—

template approach described in this article, we participated in these evaluations. The problem domain is the translation of Chinese news text into English. Table 11 gives an overview on the training and test data. The English vocabulary consists of full-form words that have been converted to lowercase letters. The number of sentences has been artificially increased by adding certain parts of the original training material more than once to the training corpus, in order to give larger weight to those parts of the training corpus that consist of high-quality aligned Chinese news text and are therefore expected to be especially helpful for the translation of the test data.

Table 12
Results of Chinese–English NIST MT evaluation, June 2002, large data track (NIST-09 score: larger values are better).

System	NIST-09 score
Alignment template approach	7.65
Competing research systems	5.03–7.34
Best of six commercial off-the-shelf systems	6.08

The Chinese language poses special problems because the boundaries of Chinese words are not marked. Chinese text is provided as a sequence of characters, and it is unclear which characters have to be grouped together to obtain entities that can be interpreted as words. For statistical MT, it would be possible to ignore this fact and treat the Chinese characters as elementary units and translate them into English. Yet preliminary experiments showed that the existing alignment models produce better results if the Chinese characters are segmented in a preprocessing step into single words. We use the LDC segmentation tool.[5]

For the English corpus, the following preprocessing steps are applied. First, the corpus is tokenized; it is then segmented into sentences, and all uppercase characters are converted to lowercase. As the final evaluation criterion does not distinguish case, it is not necessary to deal with the case information.

Then, the preprocessed Chinese and English corpora are sentence aligned in which the lengths of the source and target sentences are significantly different. From the resulting corpus, we automatically replace translations. In addition, only sentences with less than 60 words in English and Chinese are used.

To improve the translation of Chinese numbers, we use a categorization of Chinese number and date expressions. For the statistical learning, all number and date expressions are replaced with one of two generic symbols, *$number* or *$date*. The number and date expressions are subjected to a rule-based translation by simple lexicon lookup. The translation of the number and date expressions is inserted into the output using the alignment information. For Chinese and English, this categorization is implemented independently of the other language.

To evaluate MT quality on this task, NIST made available the NIST-09 evaluation tool. This tool provides a modified BLEU score by computing a weighted precision of n-grams modified by a length penalty for very short translations. Table 12 shows the results of the official evaluation performed by NIST in June 2002. With a score of 7.65, the results obtained were statistically significantly better than any other competing approach. Differences in the NIST score larger than 0.12 are statistically significant at the 95% level. We conclude that the developed alignment template approach is also applicable to unrelated language pairs such as Chinese–English and that the developed statistical models indeed seem to be largely language-independent. Table 13 shows various example translations.

7. Conclusions

We have presented a framework for statistical MT for natural languages which is more general than the widely used source–channel approach. It allows a baseline MT

5 The LDC segmentation tool is available at http://morph.ldc.upenn.edu/Projects/Chinese/LDC_ch.htm#cseg.

Table 13
Example translations for Chinese–English MT.

Reference	Significant Accomplishment Achieved in the Economic Construction of the Fourteen Open Border Cities in China
Translation	The opening up of the economy of China's fourteen City made significant achievements in construction
Reference	Xinhua News Agency, Beijing, Feb. 12—Exciting accomplishment has been achieved in 1995 in the economic construction of China's fourteen border cities open to foreigners.
Translation	Xinhua News Agency, Beijing, February 12—China's opening up to the outside world of the 1995 in the fourteen border pleased to obtain the construction of the economy.
Reference	Foreign Investment in Jiangsu's Agriculture on the Increase
Translation	To increase the operation of foreign investment in Jiangsu agriculture
Reference	According to the data provided today by the Ministry of Foreign Trade and Economic Cooperation, as of November this year, China has actually utilized 46.959 billion US dollars of foreign capital, including 40.007 billion US dollars of direct investment from foreign businessmen.
Translation	The external economic and trade cooperation Department today provided that this year, the foreign capital actually utilized by China on November to US $46.959 billion, including of foreign company direct investment was US $40.007 billion.
Reference	According to officials from the Provincial Department of Agriculture and Forestry of Jiangsu, the "Three-Capital" ventures approved by agencies within the agricultural system of Jiangsu Province since 1994 have numbered more than 500 and have utilized over 700 million US dollars worth of foreign capital, respectively three times and seven times more than in 1993.
Translation	Jiangsu Province for the Secretaries said that, from the 1994 years, Jiangsu Province system the approval of the "three-funded" enterprises, there are more than 500, foreign investment utilization rate of more than US $700 million, 1993 years before three and seven.
Reference	The actual amount of foreign capital has also increased more than 30% as compared with the same period last year.
Translation	The actual amount of foreign investment has increased by more than 30% compared with the same period last year.
Reference	Import and Export in Pudong New District Exceeding 9 billion US dollars This Year
Translation	Foreign trade imports and exports of this year to the Pudong new Region exceeds US $9 billion

system to be extended easily by adding new feature functions. We have described the alignment template approach for statistical machine translation, which uses two different alignment levels: a phrase-level alignment between phrases and a word-level alignment between single words. As a result the context of words has a greater influence, and the changes in word order from source to target language can be learned explicitly. An advantage of this method is that machine translation is learned fully automatically through the use of a bilingual training corpus. We have shown that the presented approach is capable of achieving better translation results on various tasks compared to other statistical, example-based, or rule-based translation systems. This is especially interesting, as our system is structured simpler than many competing systems.

We expect that better translation can be achieved by using models that go beyond the flat phrase segmentation that we perform in our model. A promising avenue is to gradually extend the model to take into account to some extent the recursive structure of natural languages using ideas from Wu and Wong (1998) or Alshawi, Bangalore, and Douglas (2000). We expect other improvements as well from learning nonconsecutive phrases in source or target language and from better generalization methods for the learned-phrase pairs.

Acknowledgments

The work reported here was carried out while the first author was with the Lehrstuhl für Informatik VI, Computer Science Department, RWTH Aachen–University of Technology.

References

Alshawi, Hiyan, Srinivas Bangalore, and Shona Douglas. 2000. Learning dependency translation models as collections of finite state head transducers. *Computational Linguistics*, 26(1):45–60.

Bellman, Richard. 1957. *Dynamic Programming*. Princeton University Press, Princeton.

Berger, Adam L., Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, John R. Gillett, John D. Lafferty, Harry Printz, and Lubos Ureš. 1994. The Candide system for machine translation. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 157–162, Plainsboro, NJ, March.

Berger, Adam L., Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72.

Brown, Peter F., J. Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Charniak, Eugene, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for machine translation. In *MT Summit IX*, pages 40–46, New Orleans, September.

Darroch, J. N. and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:1470–1480.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–22.

García-Varea, Ismael, Francisco Casacuberta, and Hermann Ney. 1998. An iterative, DP-based search algorithm for statistical machine translation. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'98)*, pages 1235–1238, Sydney, November.

García-Varea, Ismael, Franz Josef Och, Hermann Ney, and Francisco Casacuberta. 2001. Refined lexicon models for statistical machine translation using a maximum entropy approach. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 204–211, Toulouse, France, July.

Germann, Ulrich, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2001. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 228–235, Toulouse, France, July.

Gildea, Daniel. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 80–87, Sapporo, Japan, July.

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

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, pages 127–133, Edmonton, Alberta.

Marcu, Daniel and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pages 133–139, Philadelphia, July.

Ney, Hermann. 1995. On the probabilistic-interpretation of neural-network classifiers and discriminative training criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):107–119.

Ney, Hermann, Margit Generet, and Frank Wessel. 1995. Extensions of absolute discounting for language modeling. In *Proceedings of the Fourth European Conference on Speech Communication and Technology*, pages 1245–1248, Madrid, September.

Ney, Hermann, Franz Josef Och, and Stephan Vogel. 2001. The RWTH system for statistical translation of spoken dialogues. In *Proceedings of the ARPA Workshop on Human Language Technology*, San Diego, March.

Nießen, Sonja, Franz Josef Och, Gregor Leusch, and Hermann Ney. 2000. An evaluation tool for machine translation: Fast evaluation for machine translation research. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC)*, pages 39–45, Athens, May.

Nießen, Sonja, Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1998. A DP-based search algorithm for statistical machine translation. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 960–967, Montreal, August.

Och, Franz Josef. 1999. An efficient method for determining bilingual word classes. In *EACL '99: Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 71–76, Bergen, Norway, June.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.

Och, Franz Josef and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, July.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, Franz Josef, Nicola Ueffing, and Hermann Ney. 2001. An efficient A* search algorithm for statistical machine translation. In *Data-Driven Machine Translation Workshop*, pages 55–62, Toulouse, France, July.

Papineni, Kishore A., Salim Roukos, and R. Todd Ward. 1997. Feature-based language understanding. In *European Conference on Speech Communication and Technology*, pages 1435–1438, Rhodes, Greece, September.

Papineni, Kishore A., Salim Roukos, and R. Todd Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 189–192, Seattle, May.

Papineni, Kishore A., Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY.

Steinbiss, Volker, Bach-Hiep Tran, and Hermann Ney. 1994. Improvements in beam search. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'94)*, pages 2143–2146, Yokohama, Japan, September.

Tessiore, Lorenzo and Walther von Hahn. 2000. Functional validation of a machine interpretation system: Verbmobil. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translations*, pages 611–631. Springer, Berlin.

Tillmann, Christoph. 2001. *Word Re-ordering and Dynamic Programming Based Search Algorithms for Statistical Machine Translation*. Ph.D. thesis, Computer Science Department, RWTH Aachen, Germany.

Tillmann, Christoph. 2003. A projection extension algorithm for statistical machine translation. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Sapporo, Japan.

Tillmann, Christoph and Hermann Ney. 2000. Word re-ordering and DP-based search in statistical machine translation. In *COLING '00: The 18th International Conference on Computational Linguistics*, pages 850–856, Saarbrücken, Germany, July.

Tillmann, Christoph and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133.

Tillmann, Christoph, Stephan Vogel, Hermann Ney, and Alex Zubiaga. 1997. A DP-based search using monotone alignments in statistical translation. In *Proceedings of the 35th Annual Conference of the Association for Computational Linguistics*, pages 289–296, Madrid, July.

Venugopal, Ashish, Stephan Vogel, and Alex Waibel. 2003. Effective phrase translation extraction from alignment models. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 319–326, Sapporo, Japan, July.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING '96: The 16th International Conference on Computational Linguistics*, pages 836–841, Copenhagen, August.

Wahlster, Wolfgang, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translations*. Springer, Berlin.

Wang, Ye-Yi. 1998. *Grammar Inference and Statistical Machine Translation*. Ph.D. thesis, School of Computer Science, Language Technologies Institute, Carnegie Mellon University, Pittsburgh.

Wang, Ye-Yi and Alex Waibel. 1997. Decoding algorithm in statistical translation. In *Proceedings of the 35th Annual Conference of the Association for Computational Linguistics*, pages 366–372, Madrid, July.

Wu, Dekai and William Wong. 1998. Machine translation with a stochastic grammatical channel. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 1408–1414, Montreal, August.

Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 523–530, Toulouse, France, July.