Improved Alignment Models for Statistical Machine Translation

Franz Josef Och, Christoph Tillmann, and Hermann Ney
Lehrstuhl für Informatik VI
RWTH Aachen - University of Technology
Ahornstraße 55; 52056 Aachen; GERMANY
{och,tillmann,ney}@informatik.rwth-aachen.de

Abstract

In this paper, we describe improved alignment models for statistical machine translation. The statistical translation approach uses two types of information: a translation model and a language model. The language model used is a bigram or general m-gram model. The translation model is decomposed into a lexical and an alignment model. We describe two different approaches for statistical translation and present experimental results. The first approach is based on dependencies between single words, the second approach explicitly takes shallow phrase structures into account, using two different alignment levels: a phrase level alignment between phrases and a word level alignment between single words. We present results using the Verbmobil task (German-English, 6000-word vocabulary) which is a limited-domain spoken-language task. The experimental tests were performed on both the text transcription and the speech recognizer output.

1   Statistical Machine Translation

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

$$
\hat{e}_1^I = \arg\max_{e_1^I} \{Pr(e_1^I | f_1^J)\}
$$

$$
= \arg\max_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)\} \quad . (1)
$$

The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. $Pr(e_1^I)$ is the *language model* of the target language, whereas $Pr(f_1^J | e_1^I)$ is the *translation model*.

Many statistical translation models (Vogel et al., 1996; Tillmann et al., 1997; Niessen et al., 1998; Brown et al., 1993) try to model word-to-word correspondences between source and target words. The model is often further restricted that each source word is assigned *exactly one* target word. These alignment models are similar to the concept of Hidden Markov models (HMM) in speech recognition. The alignment mapping is $j \rightarrow i = a_j$ from source position j to target position $i = a_j$. The use of this alignment model raises major problems as it fails to capture dependencies between groups of words. As experiments have shown it is difficult to handle different word order and the translation of compound nouns.

In this paper, we will describe two methods for statistical machine translation extending the baseline alignment model in order to account for these problems. In section 2, we shortly review the single-word based approach described in (Tillmann et al., 1997) with some recently implemented extensions allowing for one-to-many alignments. In section 3 we describe the alignment template approach which explicitly models shallow phrases and in doing so tries to overcome the above mentioned restrictions of single-word alignments. The described method is an improvement of (Och and Weber, 1998), resulting in an improved training and a faster search organization. The basic idea is to model two different alignment levels: a phrase level alignment between phrases and a word level alignment between single words within these phrases. Similar aims are pursued by (Alshawi et al., 1998; Wang and Waibel, 1998) but differently approached. In section 4 we compare the two methods using the Verbmobil task.

2 Single-Word Based Approach

2.1 Basic Approach

In this section, we shortly review a translation approach based on the so-called *monotonicity* requirement (Tillmann et al., 1997). Our aim is to provide a basis for comparing the two different translation approaches presented.

In Eq. (1), $Pr(e_1^I)$ is the language model, which is a trigram language model in this case. For the translation model $Pr(f_1^J|e_1^I)$ we make the assumption that each source word is aligned to exactly one target word (a relaxation of this assumption is described in section 2.2). For our model, the probability of alignment a_j for position j depends on the previous alignment position a_{j-1} (Vogel et al., 1996). Using this assumption, there are two types of probabilities: the alignment probabilities denoted by $p(a_j|a_{j-1})$ and the lexicon probabilities denoted by $p(f_j|e_{a_j})$. The string translation probability can be re-written:

$$Pr(f_1^J|e_1^I) \quad = \quad \sum_{a_1^J} \prod_j [p(a_j|a_{j-1}) \cdot p(f_j|e_{a_j})]$$

For the training of the above model parameters, we use the maximum likelihood criterion in the so-called maximum approximation. When aligning the words in parallel texts (for Indo-European language pairs like Spanish-English, French-English, Italian-German,...), we typically observe a strong localization effect. In many cases, although not always, there is an even stronger restriction: over large portions of the source string, the alignment is monotone. In this approach, we first assume that the alignments satisfy the *monotonicity* requirement. Within the translation search, we will introduce suitably restricted permutations of the source string, to satisfy this requirement. For the alignment model, the monotonicity property allows only transitions from a_{j-1} to a_j with a jump width δ: $\delta \equiv a_j - a_{j-1} \in \{0, 1, 2\}$. Theses jumps correspond to the following three cases ($\delta = 0, 1, 2$):

- $\delta = 0$ (horizontal transition = alignment repetition): This case corresponds to a target word with two or more aligned source words.

- $\delta = 1$ (forward transition = regular alignment): This case is the regular one: a single

new target word is generated.

- $\delta = 2$ (skip transition = non-aligned word): This case corresponds to skipping a word, i.e. there is a word in the target string with no aligned word in the source string.

The possible alignments using the monotonicity assumption are illustrated in Fig. 1. Monotone alignments are paths through this uniform trellis structure. Using the concept of
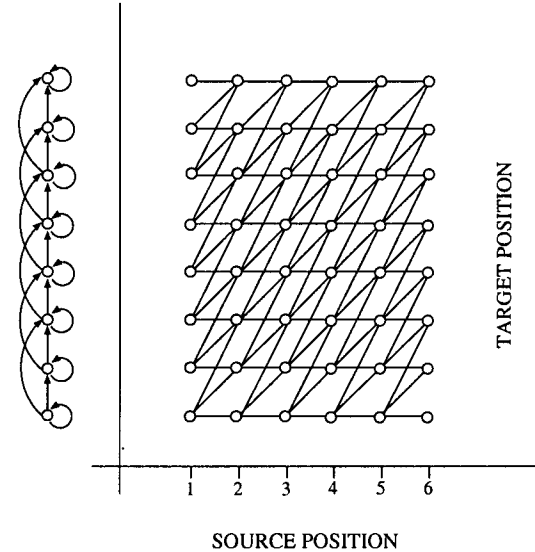

Figure 1: Illustration of alignments for the monotone HMM.

monotone alignments a search procedure can be formulated which is equivalent to finding the best path through a translation lattice, where the following auxiliary quantity is evaluated using dynamic programming: Here, e and e' are

$Q_{e'}(j, e)$ probability of the best partial hypothesis (e_1^i, a_1^j) with $e_i = e$, $e_{i-1} = e'$ and $a_j = i$.

the two final words of the hypothesized target string. The auxiliary quantity is evaluated in a position-synchronous way, where j is the processed position in the source string. The result of this search is a mapping: $j \rightarrow (a_j, e_{a_j})$, where each source word is mapped to a target position a_j and a word e_{a_j} at this position. For a trigram language model the following DP recursion equation is evaluated:

$$Q_{e'}(j, e) = p(f_j|e) \cdot \max\{$$
$$p(0) \cdot Q_{e'}(j-1, e),$$
$$p(1) \cdot \max_{e''}\{p(e|e', e'') \cdot Q_{e''}(j-1, e')\}$$
$$p(2) \cdot \max_{e'',e'''}\{p(e|e', e'') \cdot p(e'|e'', e''')$$
$$\cdot Q_{e'''}(j-1, e'')\}\}$$

$p(\delta)$ is the alignment probability for the three cases above, $p(\cdot|\cdot,\cdot)$ denoting the trigram language model. e, e', e'', e''' are the four final words which are considered in the dynamic programming taking into account the monotonicity restriction and a trigram language model. The DP equation is evaluated recursively to find the best partial path to each grid point (j, e', e). No explicit length model for the length of the generated target string e_1^I given the source string f_1^J is used during the generation process. The length model is implicitly given by the alignment probabilities. The optimal translation is obtained by carrying out the following optimization:

$$\max_{e', e}\{Q_{e'}(J, e) \cdot p(\$|e, e')\},$$

where J is the length of the input sentence and $\$$ is a symbol denoting the sentence end. The complexity of the algorithm for full search is $J \cdot E^4$, where E is the size of the target language vocabulary. However, this is drastically reduced by beam-search.

2.2 One-to-many alignment model

The baseline alignment model does not permit that a source word is aligned with two or more target words. Therefore, lexical correspondences like 'Zahnarzttermin' for *dentist's appointment* cause problems because a single source word must be mapped on two or more target words. To solve this problem for the alignment in training, we first reverse the translation direction, i. e. English is now the source language, and German is the target language. For this reversed translation direction, we perform the usual training and then check the alignment paths obtained in the maximum approximation. Whenever a German word is aligned with a sequence of the adjacent English words, this sequence is added to the English vocabulary as an additional entry. As a result, we have an extended English vocabulary. Using this new vocabulary, we then perform the stan-

dard training for the original translation direction.

2.3 Extension to Handle Non-Monotonicity

Our approach assumes that the alignment is monotone with respect to the word order for the lion's share of all word alignments. For the translation direction German-English the monotonicity constraint is violated mainly with respect to the verb group. In German, the verb group usually consists of a left and a right verbal brace, whereas in English the words of the verb group usually form a sequence of consecutive words. For our DP search, we use a left-to-right beam-search concept having been introduced in speech recognition, where we rely on beam-search as an efficient pruning technique in order to handle potentially huge search spaces. Our ultimate goal is speech translation aiming at a tight integration of speech recognition and translation (Ney, 1999). The results presented were obtained by using a quasi-monotone search procedure, which proceeds from left to right along the position of the source sentence but allows for a small number of source positions that are not processed monotonically. The word re-orderings of the source sentence positions were restricted to the words of the German verb group. Details of this approach will be presented elsewhere.

3 Alignment Template Approach

A general deficiency of the baseline alignment models is that they are only able to model correspondences between single words. A first countermeasure was the refined alignment model described in section 2.2. A more systematic approach is to consider whole phrases rather than single words as the basis for the alignment models. In other words, a whole group of adjacent words in the source sentence may be aligned with a whole group of adjacent words in the target language. 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.

3.1 The word level alignment: alignment templates

In this section we will describe how we model the translation of shallow phrases.

T3 · · ■ ■ ■
T2 · ■ · · ·
T1 ■ · · · ·
　　S1 S2 S3 S4 S5

T1: zwei, drei, vier, fünf, ...
T2: Uhr
T3: vormittags, nachmittags, abends, ...

S1: two, three, four, five, ...
S2: o'clock
S3: in
S4: the
S5: morning, evening, afternoon, ...

Figure 2: Example of an alignment template and bilingual word classes.

The key element of our translation model are the *alignment templates.* An alignment template z is a triple $(\tilde{F}, \tilde{E}, \tilde{A})$ which describes the alignment $\tilde{A}$ between a source class sequence $\tilde{F}$ and a target class sequence $\tilde{E}$.

The alignment $\tilde{A}$ is represented as a matrix with binary values. 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 to a target word then it is aligned to the empty word e_0 which shall be at the imaginary position $i = 0$. This alignment representation is a generalization of the baseline alignments described in (Brown et al., 1993) and allows for many-to-many alignments.

The classes used in $\tilde{F}$ and $\tilde{E}$ are automatically trained bilingual classes using the method described in (Och, 1999) and constitute a partition of the vocabulary of source and target language. The class functions $\mathcal{F}$ and $\mathcal{E}$ map words to their classes. The use of classes instead of words themselves has the advantage of a better generalization. If there exist classes in source and target language which contain all towns it is possible that an alignment template learned using a special town can be generalized to all towns. In Fig. 2 an example of an alignment template is shown.

An alignment template $z = (\tilde{F}, \tilde{E}, \tilde{A})$ is ap-

plicable to a sequence of source words $\tilde{f}$ if the alignment template classes and the classes of the source words are equal: $\mathcal{F}(\tilde{f}) = \tilde{F}$. The application of the alignment template z constrains the target words $\tilde{e}$ to correspond to the target class sequence: $\mathcal{E}(\tilde{e}) = \tilde{E}$.

The application of an alignment template does not determine the target words, but only constrains them. For the selection of words from classes we use a statistical model for $p(\tilde{e}|z, \tilde{f})$ based on the lexicon probabilities of a statistical lexicon $p(f|e)$. We assume a mixture alignment between the source and target language words constrained by the alignment matrix $\tilde{A}$:

$$p(\tilde{f}|(\tilde{F}, \tilde{E}, \tilde{A}), \tilde{e}) = \delta(\mathcal{E}(\tilde{e}), \tilde{E})\delta(\mathcal{F}(\tilde{f}), \tilde{F}) \cdot$$
$$\prod_{j=1}^{I} p(f_j|\tilde{A}, \tilde{e}) \quad (2)$$

$$p(f_j|\tilde{A}, \tilde{e}) = \sum_{i=0}^{I} p(i|j; \tilde{A}) \cdot p(f_j|e_i) \quad (3)$$

$$p(i|j; \tilde{A}) = \frac{\tilde{A}(i, j)}{\sum_i \tilde{A}(i, j)} \quad (4)$$

3.2 The phrase level alignment

In order to describe the phrase level alignment 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 $(k = 1, \ldots, K)$:

$$f_1^J = \tilde{f}_1^K, \qquad \tilde{f}_k = f_{j_{k-1}+1}, \ldots, f_{j_k}$$
$$e_1^I = \tilde{e}_1^K, \qquad \tilde{e}_k = e_{i_{k-1}+1}, \ldots, e_{i_k} .$$

In order to simplify the notation and the presentation, we ignore the fact that there can be a large number of possible segmentations and assume that there is only one segmentation. In the previous section, we have described the alignment *within* the phrases. For the alignment $\tilde{a}_1^K$ *between* the source phrases $\tilde{e}_1^K$ and the target phrases $\tilde{f}_1^K$, we obtain the following equation:

$$Pr(f_1^J|e_1^I) = Pr(\tilde{f}_1^K|\tilde{e}_1^K)$$
$$= \sum_{\tilde{a}_1^K} Pr(\tilde{a}_1^K, \tilde{f}_1^K|\tilde{e}_1^K)$$
$$= \sum_{\tilde{a}_1^K} Pr(\tilde{a}_1^K|\tilde{e}_1^K) \cdot Pr(\tilde{f}_1^K|\tilde{a}_1^K, \tilde{e}_1^K)$$
$$= \sum_{\tilde{a}_1^K} \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_1^{k-1}, K) \cdot p(\tilde{f}_k|\tilde{e}_{\tilde{a}_k}) .$$

23

For the phrase level alignment we use a first-order alignment model $p(\tilde{a}_k|\tilde{a}_1^{k-1}, K) = p(\tilde{a}_k|\tilde{a}_{k-1}, K)$ which is in addition constrained to be a permutation of the K phrases.

For the translation of one phrase, we introduce the alignment template as an unknown variable:

$$p(\tilde{f}|\tilde{e}) = \sum_z p(z|\tilde{e}) \cdot p(\tilde{f}|z, \tilde{e}) \qquad (5)$$

The probability $p(z|\tilde{e})$ to apply an alignment template gets estimated by relative frequencies (see next section). The probability $p(\tilde{f}|z, \tilde{e})$ is decomposed by Eq. (2).

3.3 Training

In this section we show how we obtain the parameters of our translation model by using a parallel training corpus:

1. We train two HMM alignment models (Vogel et al., 1996) for the two translation directions $f \rightarrow e$ and $e \rightarrow f$ by applying the EM-algorithm. However we do not apply maximum approximation in training, thereby obtaining slightly improved alignments.

2. For each translation direction we calculate the Viterbi-alignment of the translation models determined in the previous step. Thus we get two alignment vectors a_1^J and b_1^I for each sentence.

   We increase the quality of the alignments by combining the two alignment vectors into one alignment matrix using the following method. $A_1 = \{(a_j, j)|j = 1 \ldots J\}$ and $A_2 = \{(i, b_i)|i = 1 \ldots I\}$ denote the set of links in the two Viterbi-alignments. In a first step the intersection $A = A_1 \cap A_2$ is determined. The elements within A are justified by both Viterbi-alignments and are therefore very reliable. We now extend the alignment A iteratively by adding links (i, j) occurring only in A_1 or in A_2 if they have a neighbouring link already in A or if neither the word f_j nor the word e_i are aligned in A. The alignment (i, j) has the neighbouring links $(i - 1, j)$, $(i, j - 1)$, $(i + 1, j)$, and $(i, j + 1)$. In the Verbmobil task (Table 1) the precision of the baseline Viterbi alignments is 83.3 percent with English as source language and 81.8 percent with German as source language. Using this heuristic we get an alignment matrix with a precision of 88.4 percent without loss in recall.

3. We estimate a bilingual word lexicon $p(f|e)$ by the relative frequencies of the alignment determined in the previous step:

$$p(f|e) = \frac{n_A(f, e)}{n(e)} \qquad (6)$$

Here $n_A(f, e)$ is the frequency that the word f is aligned to e and $n(e)$ is the frequency of e in the training corpus.

4. We determine word classes for source and target language. A naive approach for doing this would be the use of monolingually optimized word classes in source and target language. Unfortunately we can not expect that there is a direct correspondence between independently optimized classes. Therefore monolingually optimized word classes do not seem to be useful for machine translation.

   We determine correlated bilingual classes by using the method described in (Och, 1999). The basic idea of this method is to apply a maximum-likelihood approach to the joint probability of the parallel training corpus. The resulting optimization criterion for the bilingual word classes is similar to the one used in monolingual maximum-likelihood word clustering.

5. We count all phrase-pairs of the training corpus which are consistent with the alignment matrix determined in step 2. A phrase-pair is consistent with the alignment if the words within the source phrase are only aligned to words within the target 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 needed by Eq. (5) is estimated by relative frequency:

$$p(z = (\tilde{F}, \tilde{E}, \tilde{A})|\tilde{e}) = \frac{n(z) \cdot \delta(\tilde{E}, \mathcal{E}(\tilde{e}))}{n(\mathcal{E}(\tilde{e}))} \qquad (7)$$

Fig. 3 shows some of the extracted alignment templates. The extraction algorithm

?
afternoon
the
in
o'clock
two
,
maybe
at
nineteenth
the
about
how
,
okay

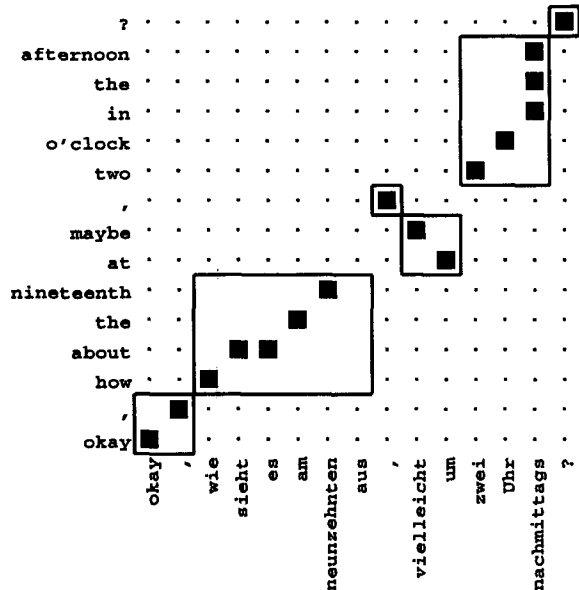okay , wie sieht es am neunzehnten aus ' vielleicht um zwei Uhr nachmittags ?

Figure 3: Example of a word alignment and some learned alignment templates.

does not perform a selection of good or bad alignment templates - it simply extracts all possible alignment templates.

3.4 Search

For decoding we use the following search criterion:

$$\arg\max_{e_1^I} \{p(e_1^I) \cdot p(e_1^I | f_1^J)\} \qquad (8)$$

This decision rule is an approximation to Eq. (1) which would use the translation probability $p(f_1^J | e_1^I)$. Using the simplification it is easy to integrate translation and language model in the search process as both models predict target words. As experiments have shown this simplification does not affect the quality of translation results.

To allow the influence of long contexts we use a class-based five-gram language model with backing-off.

The search space denoted by Eq. (8) is very large. Therefore we apply two preprocessing steps before the translation of a sentence:

1. 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 to a sub-sequence of the source sentence is called *alignment template instantiation.*

2. We now perform a segmentation of the input sentence. We search for a sequence of phrases $\tilde{f}_1 \circ \cdots \circ \tilde{f}_k = f_1^J$ with:

$$\arg\max_{\tilde{f}_1 \circ \cdots \circ \tilde{f}_k = f_1^J} \prod_{k=1}^{K} \max_z p(z | \tilde{f}_k) \qquad (9)$$

This is done efficiently by dynamic programming. Because of the simplified decision rule (Eq. (8)) it is used in Eq. (9) $p(z | \tilde{f}_k)$ instead of $p(z | \tilde{e}_k)$.

Afterwards the actual translation process begins. It has a search organization along the positions of the target language string. In search we produce partial hypotheses, each of which contains the following information:

1. the last target word produced,

2. the state of the language model (the classes of the last four target words),

3. a bit-vector representing the already covered positions of the source sentence,

4. a reference to the alignment template instantiation which produced the last target word,

5. the position of the last target word in the alignment template instantiation,

6. the accumulated costs (the negative logarithm of the probabilities) of all previous decisions,

7. a reference to the previous partial hypothesis.

A partial hypothesis is extended by appending one target word. The set of all partial hypotheses can be structured as a graph with a source node representing the sentence start, leaf nodes representing full translations and intermediate nodes representing partial hypotheses. We recombine partial hypotheses which cannot be distinguished by neither language model nor translation model. When the elements 1 - 5 of two partial hypotheses do not allow to distinguish between two hypotheses it is possible to drop the hypothesis with higher costs for the subsequent search process.

We also use beam-search in order to handle the huge search space. We compare in beam-search hypotheses which cover different parts of

the input sentence. This makes the comparison of the costs somewhat problematic. Therefore we integrate an (optimistic) estimation of the remaining costs to arrive at a full translation. This can be done efficiently by determining in advance for each word in the source language sentence a lower bound for the costs of the translation of this word. Together with the bit-vector stored in a partial hypothesis it is possible to achieve an efficient estimation of the remaining costs.

4 Translation results

The "Verbmobil Task" (Wahlster, 1993) is a speech translation task in the domain of appointment scheduling, travel planning, and hotel reservation. The task is difficult because it consists of spontaneous speech and the syntactic structures of the sentences are less restricted and highly variable.

The translation direction is from German to English which poses special problems due to the big difference in the word order of the two languages. We present results on both the text transcription and the speech recognizer output using the alignment template approach and the single-word based approach.

The text input was obtained by manually transcribing the spontaneously spoken sentences. There was no constraint on the length of the sentences, and some of the sentences in the test corpus contain more than 50 words. Therefore, for text input, each sentence is split into shorter units using the punctuation marks. The segments thus obtained were translated separately, and the final translation was obtained by concatenation.

In the case of speech input, the speech recognizer along with a prosodic module produced so-called prosodic markers which are equivalent to punctuation marks in written language. The experiments for speech input were performed on the single-best sentence of the recognizer. The recognizer had a word error rate of 31.0%. Considering only the real words without the punctuation marks, the word error rate was smaller, namely 20.3%.

A summary of the corpus used in the experiments is given in Table 1. Here the term word refers to full-form word as there is no morphological processing involved. In some of our ex-

periments we use a domain-specific preprocessing which consists of a list of 803 (for German) and 458 (for English) word-joinings and word-splittings for word compounds, numbers, dates and proper names. To improve the lexicon probabilities and to account for unseen words we added a manually created German-English dictionary with 13 388 entries. The classes used were constrained so that all proper names were included in a single class. Apart from this, the classes were automatically trained using the described bilingual clustering method. For each of the two languages 400 classes were used.

For the single-word based approach, we used the manual dictionary as well as the preprocessing steps described above. Neither the translation model nor the language model used classes in this case. In principal, when re-ordering words of the source string, words of the German verb group could be moved over punctuation marks, although it was penalized by a constant cost.

Table 1: Training and test conditions for the Verbmobil task. The extended vocabulary includes the words of the manual dictionary. The trigram perplexity (PP) is given.

		German	English
Train	Sentences	34 465	
	Words	363 514	383 509
	Voc.	6 381	3 766
	Extended Voc.	9 062	8 437
Test	Sentences	147	
	Words	1 968	2 173
	PP	–	31.5

In all experiments, we use the following three error criteria:

- WER (word error rate):
  The WER is computed as the minimum number of substitution, insertion and deletion operations that have to be performed to convert the generated string into the target string. This performance criterion is widely used in speech recognition.

- PER (position-independent word error rate):
  A shortcoming of the WER is the fact that it requires a perfect word order. This is

26

Table 2: **Experiments for Text and Speech Input**: Word error rate (WER), position-independent word error rate (PER) and subjective sentence error rate (SSER) with/without pre-processing (147 sentences = 1 968 words of the Verbmobil task).

	Input	Preproc.	WER[%]	PER[%]	SSER[%]
Single-Word Based Approach	**Text**	No	53.4	38.3	35.7
		Yes	56.0	41.2	35.3
	Speech	No	67.8	50.1	54.8
		Yes	67.8	51.4	52.7
Alignment Templates	**Text**	No	49.5	35.3	31.5
		Yes	48.3	35.1	27.2
	Speech	No	63.5	45.6	52.4
		Yes	62.8	45.6	50.3

particularly a problem for the Verbmobil task, where the word order of the German-English sentence pair can be quite different. As a result, the word order of the automatically generated target sentence can be different from that of the target sentence, but nevertheless acceptable so that the WER measure alone could be misleading. In order to overcome this problem, we introduce as additional measure the position-independent word error rate (PER). This measure compares the words in the two sentences *without* taking the word order into account. Words that have no matching counterparts are counted as substitution errors. Depending on whether the translated sentence is longer or shorter than the target translation, the remaining words result in either insertion or deletion errors in addition to substitution errors. The PER is guaranteed to be less than or equal to the WER.

- SSER (subjective sentence error rate):
  For a more detailed analysis, subjective judgments by test persons are necessary. Each translated sentence was judged by a human examiner according to an error scale from 0.0 to 1.0. A score of 0.0 means that the translation is semantically and syntactically correct, a score of 0.5 means that a sentence is semantically correct but syntactically wrong and a score of 1.0 means that the sentence is semantically wrong. The human examiner was offered the translated sentences of the two approaches at the same time. As a result we expect a better possi-

bility of reproduction.

The results of the translation experiments using the single-word based approach and the alignment template approach on text input and on speech input are summarized in Table 2. The results are shown with and without the use of domain-specific preprocessing. The alignment template approach produces better translation results than the single-word based approach. From this we draw the conclusion that it is important to model word groups in source and target language. Considering the recognition word error rate of 31% the degradation of about 20% by speech input can be expected. The average translation time on an Alpha workstation for a single sentence is about one second for the alignment template approach and 30 seconds for the single-word based search procedure.

Within the Verbmobil project other translation modules based on rule-based, example-based and dialogue-act-based translation are used. We are not able to present results with these methods using our test corpus. But in the current Verbmobil prototype the preliminary evaluations show that the statistical methods produce comparable or better results than the other systems. An advantage of the system is that it is robust and always produces a translation result even if the input of the speech recognizer is quite incorrect.

5 Summary

We have described two approaches to perform statistical machine translation which extend the baseline alignment models. The single-word

based approach allows for the the possibility of one-to-many alignments. The alignment template approach 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 both methods is that they learn fully automatically by using a bilingual training corpus and are capable of achieving better translation results on a limited-domain task than other example-based or rule-based translation systems.

Acknowledgment

References

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. 1998. Automatic acquisition of hierarchical transduction models for machine translation. In *COLING-ACL '98: Annual Conf. of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, volume 1, pages 41–47, Montreal, Quebec, Canada, August.

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

Hermann Ney. 1999. Speech translation: Coupling of recognition and translation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 517–520, Phoenix, AR, March.

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

Franz Josef Och and Hans Weber. 1998. Improving statistical natural language translation with categories and rules. In *Proc. of the 35th Annual Conf. of the Association for Computational Linguistics and the 17th Int. Conf. on Computational Linguistics*, pages 985–989, Montreal, Canada, August.

Franz Josef Och. 1999. An efficient method to determine bilingual word classes. In *EACL '99: Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics*, Bergen, Norway, June.

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

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

Wolfgang Wahlster. 1993. Verbmobil: Translation of face-to-face dialogs. In *Proc of the MT Summit IV*, pages 127–135, Kobe, Japan.

Ye-Yi Wang and Alex Waibel. 1998. Modeling with structures in statistcal machine translation. In *COLING-ACL '98: Annual Conf. of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, volume 2, pages 1357–1363, Montreal, Quebec, Canada.