Sentence Segmentation Using IBM Word Alignment Model 1

Jia Xu and Richard Zens and Hermann Ney

Chair of Computer Science VI, Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
{xujia,zens,ney}@cs.rwth-aachen.de

Abstract. In statistical machine translation, word alignment models are trained on bilingual corpora. Long sentences pose severe problems: 1. the high computational requirements; 2. the poor quality of the resulting word alignment. We present a sentence-segmentation method that solves these problems by splitting long sentence pairs. Our approach uses the lexicon information to locate the optimal split point. This method is evaluated on two Chinese-English translation tasks in the news domain. We show that the segmentation of long sentences before training significantly improves the final translation quality of a state-of-the-art machine translation system. In one of the tasks, we achieve an improvement of the BLEU score of more than 20% relative.

1 Introduction

1.1 Problem Description

In a statistical machine translation system, we define a mathematical model, train the model parameters on the parallel sentence-aligned corpora and translate the test text with this model and its parameters.

In practice, many sentences in the training corpora are long. Some translation applications cannot handle a sentence whose length is larger than a predetermined value. The reasons are memory limits and the computational complexity of the algorithms. Therefore, long sentences are usually removed during the preprocessing. But even if long sentences are included, the resulting quality is usually not as good as it is for short sentences.

1.2 Comparison with Sentence Alignment

The problem of sentence segmentation is similar to the problem of sentence alignment which was investigated by (Brown et al., 1991; Chen, 1993; Moore, 2002). In the case of the sentence segmentation, we assume that the sentence pairs are aligned correctly. The tasks are to find appropriate split points and to align the subsentences. In the case of the sentence alignment, the corpus is aligned at the document level only. Here, we have to align the sentences of two documents rather than having to find appropriate split points.

1.3 State of the Art

Previous research on the sentence segmentation problem can be found in (Nevado et al., 2003), who searches for the segmentation boundaries using a dynamic programming algorithm. This technique is based on the lexicon information. However, it only allows a monotone alignment of the bilingual segmented sentences and it requires a list of manually defined anchor words.

1.4 Idea of the Method

Inspired by the phrase extraction approach (Vogel et al., 2004), we introduce a new sentence segmentation method which does not need anchor words and allows for nonmonotone alignments of the subsentences.

Here we separate a sentence pair into two subpairs with the so-called “IBM Word Alignment Model 1”. This process is done recursively over all the sub-sentences until their lengths are smaller than a given value. This simple algorithm leads to a significant improvement in translation quality and a speed-up of the training procedure.

2 Review of the Baseline Statistical Machine Translation System

2.1 Approach

In this section, we briefly review our translation system and introduce the word alignment models.

In statistical machine translation, we are given a source language ('French') sentence $f_1^J = f_1 \ldots f_j \ldots f_J$, which is to be translated into a target language ('English') sentence $e_1^I = e_1 \ldots e_i \ldots e_I$. Among all possible target language sentences, we will choose the sentence with the highest probability:

$$\begin{aligned} \hat{e}_1^I &= \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\} \\ &= \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J | e_1^I) \right\} \quad (1) \end{aligned}$$

The decomposition into two knowledge sources in Equation 1 allows an independent modeling of target language model $Pr(e_1^I)$ and translation model $Pr(f_1^J|e_1^I)$[1], known as source-channel model (Brown et al., 1993). The target language model describes the well-formedness of the target language sentence. The translation model links the source language sentence to the target language sentence.

The argmax operation denotes the search problem, i.e. the generation of the output sentence into the target language. We have to maximize over all possible target language sentences.

The translation model $Pr(f_1^J|e_1^I)$ can be further extended to a statistical alignment model with the following equation:

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

The alignment model $Pr(f_1^J, a_1^J|e_1^I)$ introduces a 'hidden' word alignment $\mathbf{a} = a_1^J$, which describes a mapping from a source position j to a target position a_j.

2.2 Alignment Models

There are different decompositions of the alignment probability $Pr(f_1^J, a_1^J|e_1^I)$.

The IBM-1 model (Brown et al., 1993) assumes that all alignments have the same probability by using a uniform distribution:

$$p(f_1^J|e_1^I) = \prod_{j=1}^{J} \left[\frac{1}{I} \sum_{i=1}^{I} p(f_j|e_i) \right] \quad (2)$$

Hence, the word order does not affect the alignment probability.

We use the IBM-1 model and the higher-order models IBM-4 (Brown et al., 1993) and Hidden-Markov model (HMM) (Vogel et al., 1996) to train the lexicon parameters $p(f_j|e_i)$. The resulting probability distribution is more concentrated than the one trained unsing the IBM-1 model only. The training software is GIZA++ (Och and Ney, 2003).

To incorporate the context into the translation model, the alignment template translation approach (Och and Ney, 2004) is applied. A dynamic programming beam search algorithm is used to generate the translation hypothesis with maximum probability.

3 Segmentation Methods

In this section, we describe the sentence segmentation algorithm in detail. The main idea is that we use the word alignment information to find the optimal split point in a sentence pair and separate it into two pairs.

To calculate the alignment probability of a segment pair, we indicate (j_1, i_1) and (j_2, i_2) as the start and end point of a segment, respectively.

$$p(f_{j_1}^{j_2}|e_{i_1}^{i_2}) = \prod_{j=j_1}^{j_2} \left[\frac{1}{i_2 - i_1 + 1} \sum_{i=i_1}^{i_2} p(f_j|e_i) \right] \quad (3)$$

3.1 Modified IBM-1 Model

We modified the standard IBM-1 model in Equation 3 in two ways for a better segmentation quality:

1. Length normalization

[1] The notational convention will be as follows: we use the symbol $Pr(\cdot)$ to denote general probability distributions with (almost) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $p(\cdot)$.

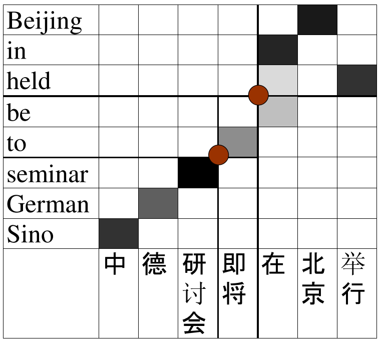

Figure 1. Sentence segmentation example.

For the sentence segmentation, a shortcoming of the simple word alignment based model is that the lengths of the separated sentence pairs are ignored. To balance the lengths of the two sub-sentence pairs, we normalize the alignment probability by the source sentence length and adjust its weight with the parameter β:

$$p_\gamma(f_{j_1}^{j_2}|e_{i_1}^{i_2}) \quad = \quad p(f_{j_1}^{j_2}|e_{i_1}^{i_2})^\gamma, \qquad (4)$$

where $\gamma = \beta \cdot \frac{1}{j_2-j_1+1} + (1-\beta)$.

2. Combination with inverse alignment model

   The standard IBM-1 model in Equation 2 calculates the conditional probability of a target sentence, given the source sentence described in Section 2. The inverse IBM-1 model means the probability of the source sentence given the target sentence. We approximate on the joint probability and combine the models in both directions:

$$p(f_{j_1}^{j_2}, e_{i_1}^{i_2}) \quad \approx \quad p(f_{j_1}^{j_2}|e_{i_1}^{i_2}) \cdot p(e_{i_1}^{i_2}|f_{j_1}^{j_2}) \qquad (5)$$

3.2 Search for Segmentation Points

As illustrated in Figure 1, we present a sentence pair as a matrix. Each position contains a lexicon probability $p(f_j|e_i)$ which is trained on the original corpora. For a clearer presentation, Figure 1 only shows a sentence pair with seven Chinese words and eight English words. The gray scale indicates the value of the probability. The darker the box, the higher the probability. All the positions are considered as possible split points.

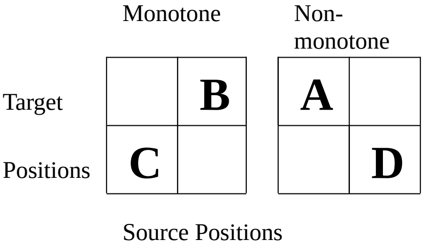

Figure 2. Two Types of Alignment

A split point (i, j) divides a matrix or a subset of the matrix into four parts, as shown in Figure 2: the upper left (A), the upper right (B), the bottom left (C) and the bottom right (D). For a segment pair with the start/end point (i_1, j_1)/(i_2, j_2), two types of alignment are possible:

1. Monotone alignment

   One case is the monotone alignment, i.e. C is combined with B. We denote this case as $\delta = 1$. The segmentation probability $p_{i,j,1}$ is the product of these two parts' alignment probabilities from Equation 5:

$$p_{i,j,1}(f_{j_1}^{j_2}, e_{i_1}^{i_2}) \quad = \quad p(f_{j_1}^{j}, e_{i_1}^{i}) \cdot p(f_{j+1}^{j_2}, e_{i+1}^{i_2})$$

2. Nonmonotone alignment

   The other case is the nonmonotone alignment indicated as $\delta = 0$, i.e. A is combined with D. We denote the probability as $p_{i,j,0}$:

$$p_{i,j,0}(f_{j_1}^{j_2}, e_{i_1}^{i_2}) \quad = \quad p(f_{j_1}^{j}, e_{i+1}^{i_2}) \cdot p(f_{j+1}^{j_2}, e_{i_1}^{i})$$

With this method, we go through all positions in the bilingual sentences and choose the split point and the orientation, which is denoted as:

$$(\hat{i}, \hat{j}, \hat{\delta}) = \underset{i,j,\delta}{\operatorname{argmax}} \left\{ p_{i,j,\delta}(f_{j_1}^{j_2}, e_{i_1}^{i_2}) \right\},$$

where $i \in [i_1, i_2 - 1]$, $j \in [j_1, j_2 - 1]$ and $\delta \in \{0, 1\}$.

To avoid the extraction of segments which are too short, e.g. single words, we use the minimum segment lengths (I_{min}, J_{min}). The possible split point is then limited to: $i \in [i_1 + I_{min} - 1, i_2 - I_{min}]$, $j \in [j_1 + J_{min} - 1, j_2 - J_{min}]$.

```
$Max = 0;$
$\forall j \in [j_1, j_2] : V_{up}[j] = \sum_{i=i_1}^{i_2} p(f_j|e_i);$
$\forall j \in [j_1, j_2] : V_{down}[j] = 0;$

$for$   $(i = i_1; i < i_2; i = i + 1)$

        $\forall_{j \in [j_1, j_2]} : V_{up}[j] = V_{up}[j] - p(f_j|e_i);$
        $\forall_{j \in [j_1, j_2]} : V_{down}[j] = V_{down}[j] + p(f_j|e_i);$

        $A = C = 1;$
        $B = \prod_{j=j_1}^{j_2} V_{up}[j];$
        $D = \prod_{j=j_1}^{j_2} V_{down}[j];$

        $for$   $(j = j_1; j < j_2; j = j + 1)$

                $A = A \cdot V_{up}[j];$
                $B = B / V_{up}[j];$
                $C = C \cdot V_{down}[j];$
                $D = D / V_{down}[j];$

                $if$      $(\max(A \cdot D, B \cdot C) > Max \wedge$
                          $i \in [i_1 + I_{min} - 1, i_2 - I_{min}] \wedge$
                          $j \in [j_1 + J_{min} - 1, j_2 - J_{min}])$
                $then$
                          $Max = \max(A \cdot D, B \cdot C);$
                          $\hat{j} = j; \hat{i} = i;$
                          $\hat{\delta} = (B \cdot C >= A \cdot D);$
```

Figure 3. Efficient Algorithm.

3.3 Efficient Algorithm

The naive implementation of the algorithm has a complexity of $O((I \cdot J)^2)$. We benefit from the structure of the IBM-1 model and calculate the alignment probability for each position using the idea of running sums/products. The complexity is reduced to $O(I \cdot J)$, i.e. factor of $100\,000$ for sentences with 100 words. But this implementation is not possible for the fertility-based higher-order models.

Details are shown in Figure 3. The input to the program are the lexicon probabilities $p(f_j|e_i)$ and the minimum sentence lengths I_{min}, J_{min}. The output are the optimal split point $(\hat{i}, \hat{j})$ and its orientation $\hat{\delta}$.

In the program, Max is the biggest alignment probability. A, B, C, D are the IBM-1 scores for each block in Figure 2. V_{up} stores the sums of the lexicon probabilities in each column in the areas A and B and V_{down} does the same for the areas C and D.

In the outer loop of the target position i, the $p(f_j|e_i)$ in the actual position is added/subtracted to/from the value in V_{down}/V_{up}, respectively. In the inner loop of the source position j, the alignment probability in the area A/B are multiplied/divided by $V_{up}[j]$, whereas the probability in C/D is multiplied/divided by the $V_{down}[j]$. After traversing all positions, the point with the maximum alignment probability is selected as the split point.

I:
中 德 研讨会 即将 | 在 北京 举行
Sino German seminar to be | held in Beijing
II:
中 德 研讨会 | 即将
Sino German seminar | to be

Figure 5. Result of the sentence segmentation example.

3.4 Recursive Segmentation

We introduce the maximum sentence lengths for the source language J_{max} and for the target language I_{max}. If a sentence is longer than the maximum length, the sentence pair is split into two sub-sentence pairs. In most cases, these sub-sentences are still too long. Therefore, the splitting is applied recursively until the length of each new sentence is less than the predefined value. The recursive algorithm is shown in Figure 4 for a bilingual sentence segmentation $S(f_1^J, e_1^I)$.

The algorithm is similar to the bracketing transduction grammars (Wu, 1997). Here, we take the local decision after each recursion. The full parsing with BTG is not feasible for long sentences because of its cubic complexity.

3.5 Segmentation Example

We take the sentence pair in Figure 1 as an example. The maximum lengths in both languages is defined as three. In practice, the segmented sentences contain from 25 to hundreds of words. Using the algorithm in Figure 4, this sentence pair is segmented as follows:

First, the lengths of the two sentences are larger than the maximum lengths, the sentences will be segmented. After the calculation with Equation 5, we find the first segmentation point: the right circle in Figure 5, i.e. $\hat{i} = 5, \hat{j} = 4$. The alignment

```
$S(f_{j_1}^{j_2}, e_{i_1}^{i_2})$ :  if   $(2 \cdot J_{min} \leq j_2 - j_1 + 1 \leq J_{max}$ and $2 \cdot I_{min} \leq i_2 - i_1 + 1 \leq I_{max})$
                   then
                        $(f_{j_1}^{j_2}, e_{i_1}^{i_2})$
                   else
                        $(\hat{i}, \hat{j}, \hat{\delta}) = \underset{i,j,\delta}{\text{argmax}} \; \{p_{i,j,\delta}(f_{j_1}^{j_2}, e_{i_1}^{i_2})\},$
                        where $i \in [i_1 + I_{min} - 1, i_2 - I_{min}], j \in [j_1 + J_{min} - 1, j_2 - J_{min}], \delta \in \{0, 1\}$

                        if     $\hat{\delta} = 1$
                        then
                               $S(f_{j_1}^{\hat{j}}, e_{i_1}^{\hat{i}}); S(f_{\hat{j}+1}^{j_2}, e_{\hat{i}+1}^{i_2})$
                        else
                               $S(f_{j_1}^{\hat{j}}, e_{\hat{i}+1}^{i_2}); S(f_{\hat{j}+1}^{j_2}, e_{i_1}^{\hat{i}})$
```

Figure 4. Recursive segmentation procedure.

is monotone, i.e. $\hat{\delta} = 1$. The result is shown in Figure 5(I).

After the first recursion, the length of the left segment in (I) is still larger than three. Hence, it is segmented again into two sub-sentence pairs shown in (II). In this case, the alignment is also monotone.

Finally, each new segment contains no more than three words.

4 Translation Experiments

4.1 Translation Tasks

We present results for two Chinese-English translation tasks. In the news domain, the corpora are provided by the Linguistic Data Consortium (LDC). Details can be found on the LDC web pages (LDC, 2003).

In the first task, the training corpus is composed of the text of a Chinese Treebank and its translation (Treebank: LDC2002E17), as well as a bilingual manual dictionary for 10K Chinese word entries and their multiple translations. This task is referred to as the “Small Data Track” in the Chinese-English DARPA TIDES evaluations carried out by NIST (NIST, 2004). In the second task, the corpus contains the articles from the Xinhua News Agency (LDC2002E18). This task has a larger vocabulary size and more named entity words.

The free parameters are optimized on the development corpus (Dev). Here, the NIST 2002 test set with 878 sentences is the development corpus, and the NIST 2004 test set with 1788 sentences is the test corpus (Test).

Table 1. Corpus Statistics

		Chinese	English
Treebank:	Sents	4 183	
	Used Sents	3 258	
	Words	115 973	128 484
	Used Words	83 081	104 675
Seg. Treebank:	Sents	14 559	
	Used Sents	10 591	
	Used Words	89 713	111 744
Xinhua:	Sents	109 792	
	Used Sents	85 130	
	Words	4 609 714	4 457 440
	Used Words	2 824 018	2 771 627
Seg. Xinhua:	Sents	612 979	
	Used Sents	427 493	
	Used Words	3 254 552	3 238 256
Lexicon:	Sents	17 832	
	Words	18 173	26 165
Dev.:	Sents	878	
	Words	26 509	23 683
Test	Sents	1 788	
	Words	55 086	52 657

4.2 Corpus Statistics

We have calculated the number of sentences (Sents) and running words (Words) in the original and segmented corpora, as shown in Table 1.

In the Treebank, there are 4 183 parallel sentences. Sentences are removed, if they are too

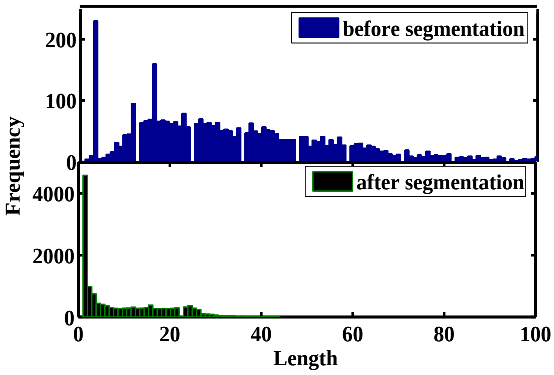

Figure 6. Histogram of the English sentence length in Treebank.

long or their source and target lengths differ too much. After this filtering, $3\,258$ sentences (Used Sents) and $83\,081$ running words (Used Words) remain. Using the sentence segmentation method, 8.0% more words are used. The average Chinese sentence length is reduced from 27.8 to 8.0.

The Xinhua corpus has longer sentences. On average, there are 42.0 words in one sentence. After segmentation, the sentence length is 7.5. The segmented corpus has 15.2% more running words used in training.

The development and test set have four references respectively, the number of running English words are their average values.

Figure 6 illustrates the histogram of the English sentence lengths in Treebank. We see that in the original corpus the sentences have very different lengths, whereas in the segmented corpus the lengths are limited to 25.

4.3 Estimation of Segmentation Parameters

Our segmentation model has two types of parameters which are optimized on development set in the task “Small Data Track”:

1. Length normalization
   Equation 4 introduces a parameter β that configures the weight of the length normalization. We used the value $\beta = 0.9$.

2. Maximum and minimum sentence lengths
   The maximum and minimum sentence lengths restrict the lengths of the sub-sentences within a range. We took the minimum lengths 1 and maximum lengths 25.

4.4 Evaluation Criteria

The commonly used criteria to evaluate the translation results in the machine translation community are:

- WER (word error rate):
  The WER is computed as the minimum number of substitution, insertion and deletion operations that have to be performed to convert the generated sentence into the reference sentence.

- PER (position-independent word error rate):
  A shortcoming of the WER is that it requires a perfect word order. The word order of an acceptable sentence can be different from that of the target sentence, so that the WER measure alone could be misleading. The PER compares the words in the two sentences ignoring the word order.

- BLEU score:
  This score measures the precision of unigrams, bigrams, trigrams and fourgrams with a penalty for too short sentences. (Papineni et al., 2002).

- NIST score:
  This score is similar to BLEU, but it uses an arithmetic average of N-gram counts rather than a geometric average, and it weights more heavily those N-grams that are more informative. (Doddington, 2002).

The BLEU and NIST scores measure accuracy, i.e. larger scores are better. In our evaluation the scores are measured as case insensitive and with respect to multiple references.

4.5 Translation Results

The evaluation is done on two tasks described in Section 4.1. In the NIST Chinese-English evaluations, the BLEU score is used as evaluation criterion. Therefore, we optimize the parameters with respect to this criterion. Using our segmentation method, we achieve a significant improvement of the BLEU score. Additinally, we obtain an improvement of the NIST score in both tasks.

We will present results of three different experiments for the “Small Data Track”:

1. **baseline**: We filter the original training corpus and use the result for training our system.

2. **filtered seg.**: We use exactly the same data that is actually used in the “baseline” experiment, but apply our splitting algorithm. Thus, the original training corpus is filtered and then split.

3. **segmented**: Here, we *first* split the training corpus and *then* apply the filtering. This enables us to use more data, because sentences that would have been removed in the “baseline” experiment are now included. Note that still some sentences are filtered out because of too different source and target lengths.

Table 2. Translation performance on the development set in “Small Data Track”.

	accuracy		error rate[%]	
method	BLEU[%]	NIST	WER	PER
baseline	15.9	6.25	74.7	48.1
filtered seg.	16.2	6.37	78.2	45.7
segmented	17.4	6.56	78.0	44.4

Table 3. Translation performance on the test set in “Small Data Track”.

	accuracy		error rate[%]	
method	BLEU[%]	NIST	WER	PER
baseline	13.5	5.80	79.1	63.8
filtered seg.	14.6	6.20	82.2	63.6
segmented	16.3	6.54	81.7	62.8

In Table 2 and Table 3, the translation results for the “Small Data Track” task are presented for the development and test set, respectively. on the development set in the “Small Data Track” task, Using the split corpora, we achieve an improvement of the BLEU score of 1.5% absolute, which is 9.4% relative. For the test set, the improvement of the BLEU score is 2.5% absolute or 20.7% relative.

In these experiments, the word error rates are worse in the “segmented” experiments, because the optimization is done for the BLEU score. Optimizing for the WER, the error rates on the development set in the baseline and the segmented experiments are almost the same, about 72%.

Table 4. Translation performance on the development set with Xinhua training corpus.

	accuracy		error rate[%]	
method	BLEU[%]	NIST	WER	PER
baseline	20.2	6.49	72.7	47.2
segmented	21.9	6.60	71.0	46.7

Table 5. Translation performance on the test set with Xinhua training corpus.

	accuracy		error rate[%]	
method	BLEU[%]	NIST	WER	PER
baseline	15.5	5.83	77.7	62.6
segmented	16.9	5.89	76.4	61.4

For the Xinhua task, shown in Table 4 and Table 5, on the development set, the BLEU score is enhanced by 1.7% absolute and by 9% relative. On the test set, the improvement of the BLEU score is 1.4% absolute or 8.4% relative.

Beside a better translation performance, using the sentence segmentation method has also other advantages:

- Enlargement of data in use
  By splitting the long sentences during the preprocessing, less words are filtered out, as shown in Table 1. Thus, we are able to use more data in the training.

- Speedup of the training process
  In the experiment of Xinhua corpus, the training with GIZA++ takes more than 10 hours. After the segmentation, it takes only about 3 hours under the same condition.

5 Discussion and Future Work

We have developed a new method to segment long bilingual sentences into several short parts using the so-called “IBM word alignment model 1”.

Experiments on the Chinese-English tasks have shown a significant improvement of the translation quality. For the Xinhua task, the BLEU score improved by about 9% relative. For the “Small Data Track” task, the improvement of the BLEU score was even more than 20% relative. Moreover, this method also enabled us to enlarge the training data in use and to speed up the training process.

Although these translation results are encouraging, we can further improve the method by considering the following cases:

- Sentence parts without translation:
  In some bilingual sentences, one or more parts of a sentence in the source or target language may have no translation at all. These parts should be marked or removed.

- Alignment of nonconsecutive sub-sentences:
  In our method we do not allow for the alignment of nonconsecutive segments.

  For example, the source sentence could be divided into three parts and the target sentence into two parts. The first and the third part of the source sentence might be translated as the first part into the target sentence, and the second part in the source sentence could be translated as the second part in the target sentence. Such a case is not yet handled here.

By solving these problems, we expect further improvements of the translation performance.

6 Acknowledgments

This work was partly funded by the DFG (Deutsche Forschungsgemeinschaft) under the grant NE572/5-1, project ”Statistische Textübersetzung” and the European Union under the integrated project TC-Star (Technology and Corpora for Speech to Speech Translation, IST-2002-FP6-506738, http://www.tc-star.org).

7 References

P. F. Brown, J. C. Lai, and R. L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proc. of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 177–184, Berkeley, California, June.

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

S. F. Chen. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proc. of the 31th Annual Meeting of the Association for Computational Linguistics*, pages 9–16, Columbus, Ohio, June.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. of Human Language Technology*, San Diego, California, March.

LDC. 2003. Linguistic data consortium resource home page. http://www.ldc.upenn.edu/Projects/TIDES.

R. C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proc. of the 5th Conf. of the Association for Machine Translation in the Americas*, pages 135–244, Tiburon, California, October.

F. Nevado, F. Casacuberta, and E. Vidal. 2003. Parallel corpora segmentation by using anchor words. In *Proc. of the EAMT/EACL Workshop on MT and Other Language Technology Tools*, pages 12–17, Budapest, Hungary, April.

NIST. 2004. Machine translation home page. http://www.nist.gov/speech/tests/mt/index.htm.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):135–244, December.

K. A. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, July.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING ’96: The 16th Int. Conf. on Computational Linguistics*, pages 836–841, Copenhagen, Denmark, August.

S. Vogel, S. Hewavitharana, M. Kolss, and A. Waibel. 2004. The ISL statistical translation system for spoken language translation. In *Proc. of the Int. Workshop on Spoken Language Translation 2004*, pages 65–72, Kyoto, Japan, September.

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