Partitioning Parallel Documents Using Binary Segmentation

Jia Xu and **Richard Zens** and **Hermann Ney**
Chair of Computer Science 6
Computer Science Department
RWTH Aachen University
D-52056 Aachen Germany
{xujia,zens,ney}@cs.rwth-aachen.de

Abstract

In statistical machine translation, large numbers of parallel sentences are required to train the model parameters. However, plenty of the bilingual language resources available on web are aligned only at the document level. To exploit this data, we have to extract the bilingual sentences from these documents.

The common method is to break the documents into segments using predefined anchor words, then these segments are aligned. This approach is not error free, incorrect alignments may decrease the translation quality.

We present an alternative approach to extract the parallel sentences by partitioning a bilingual document into two pairs. This process is performed recursively until all the sub-pairs are short enough.

In experiments on the Chinese-English FBIS data, our method was capable of producing translation results comparable to those of a state-of-the-art sentence aligner. Using a combination of the two approaches leads to better translation performance.

1 Introduction

Current statistical machine translation systems use bilingual sentences to train the parameters of the translation models. The exploitation of more bilingual sentences automatically and accurately as well as the use of these data with the limited computational requirements become crucial problems.

The conventional method for producing parallel sentences is to break the documents into sentences and to align these sentences using dynamic programming. Previous investigations can be found in works such as (Gale and Church, 1993) and (Ma, 2006). A disadvantage is that only the monotone sentence alignments are allowed.

Another approach is the binary segmentation method described in (Simard and Langlais, 2003), (Xu et al., 2005) and (Deng et al., 2006), which separates a long sentence pair into two sub-pairs recursively. The binary reordering in alignment is allowed but the segmentation decision is only optimum in each recursion step.

Hence, a combination of both methods is expected to produce a more satisfying result. (Deng et al., 2006) performs a two-stage procedure. The documents are first aligned at level using dynamic programming, the initial alignments are then refined to produce shorter segments using binary segmentation. But on the Chinese-English FBIS training corpus, the alignment accuracy and recall are lower than with Champollion (Ma, 2006).

We refine the model in (Xu et al., 2005) using a log-linar combination of different feature functions and combine it with the approach of (Ma, 2006). Here the corpora produced using both approaches are concatenated, and each corpus is assigned a weight. During the training of the word alignment models, the counts of the lexicon entries

Proceedings of the Workshop on Statistical Machine Translation, pages 78–85,
New York City, June 2006. ©2006 Association for Computational Linguistics

are linear interpolated using the corpus weights. In the experiments on the Chinese-English FBIS corpus the translation performance is improved by 0.4% of the BLEU score compared to the performance only with Champollion.

The remainder of this paper is structured as follows: First we will briefly review the baseline statistical machine translation system in Section 2. Then, in Section 3, we will describe the refined binary segmentation method. In Section 4.1, we will introduce the methods to extract bilingual sentences from document aligned texts. The experimental results will be presented in Section 4.

2 Review of the Baseline Statistical Machine Translation System

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 sentence $f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$. Among all possible target language sentences, we will choose the sentence with the highest probability:

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

The decomposition into two knowledge sources in Equation 1 allows independent modeling of target language model $Pr(e_1^I)$ and translation model $Pr(f_1^J|e_1^I)$[1]. The translation model 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.

[1]The notational convention will be 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 $p(\cdot)$.

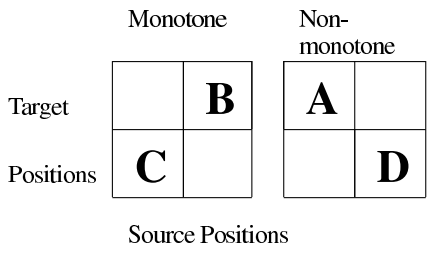

Figure 1: Two Types of Alignment

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

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

We use the IBM-1 to train the lexicon parameters $p(f|e)$, the training software is GIZA++ (Och and Ney, 2003).

To incorporate the context into the translation model, the phrase-based translation approach (Zens et al., 2005) is applied. Pairs of source and target language phrases are extracted from the bilingual training corpus and a beam search algorithm is implemented to generate the translation hypothesis with maximum probability.

3 Binary Segmentation Method

3.1 Approach

Here a document or sentence pair (f_1^J, e_1^I) [2] is represented as a matrix. Every element in the matrix contains a lexicon probability $p(f_j|e_i)$, which is trained on the original parallel corpora. Each position divides a matrix into four parts as shown in Figure 1: the bottom left (C), the upper left (A), the bottom right (D) and the upper right (B). We use m to denote the alignment direction, $m = 1$ means that the alignment is monotone, i.e. the bottom left part is connected with the upper right part, and $m = 0$ means the alignment is non-monotone, i.e. the upper left part is connected with the bottom right part, as shown in Figure 1.

3.2 Log-Linear Model

We use a log-linear interpolation to combine different models: the IBM-1, the inverse IBM-1, the an-

[2]Sentences are equivalent to segments in this paper.

chor words model as well as the IBM-4. K denotes the total number of models.

We go through all positions in the bilingual sentences and find the best position for segmenting the sentence:

$$(\hat{i},\hat{j},\hat{m}) = \underset{i,j,m}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \lambda_k h_k(j,i,m|f_1^J,e_1^I) \right\},$$

where $i \in [1, I-1]$ and $j \in [1, J-1]$ are positions in the source and target sentences respectively. The feature functions are described in the following sections. In most cases, the sentence pairs are quite long and even after one segmentation we may still have long sub-segments. Therefore, we separate the sub-segment pairs recursively until the length of each new segment is less than a defined value.

3.3 Normalized IBM-1

The function in Equation 2 can be normalized by the source sentence length with a weighting β as described in (Xu et al., 2005):

The monotone alignment is calculated as

$$\begin{aligned} h_1(j,i,1|f_1^J,e_1^I) &= \log(p(f_1^j|e_1^i)^{\beta\cdot\frac{1}{j}+(1-\beta)} \\ &\cdot p(f_{j+1}^J|e_{i+1}^I)^{\beta\cdot\frac{1}{J-j}+(1-\beta)}), \end{aligned} \quad (3)$$

and the non-monotone alignment is formulated in the same way.

We also use the inverse IBM-1 as a feature, by exchanging the place of e_1^i and f_1^j its monotone alignment is calculated as:

$$\begin{aligned} h_2(j,i,1|f_1^J,e_1^I) &= \log(p(e_1^i|f_1^j)^{\beta\cdot\frac{1}{i}+(1-\beta)} \\ &\cdot p(e_{i+1}^I|f_{j+1}^J)^{\beta\cdot\frac{1}{I-i}+(1-\beta)}) \end{aligned} \quad (4)$$

3.4 Anchor Words

In the task of extracting parallel sentences from the paragraph-aligned corpus, selecting some anchor words as preferred segmentation positions can effectively avoid the extraction of incomplete segment pairs. Therefore we use an anchor words model to prefer the segmentation at the punctuation marks, where the source and target words are identical:

$$h_3(j,i,m|f_1^J,e_1^I) = \begin{cases} 1 : f_j = e_i \wedge e_i \in \mathcal{A} \\ 0 : \text{otherwise} \end{cases}$$

$\mathcal{A}$ is a user defined anchor word list, here we use $\mathcal{A}$={.,"?;}. If the corresponding model scaling factor λ_3 is assigned a high value, the segmentation positions are mostly after anchor words.

3.5 IBM-4 Word Alignment

If we already have the IBM-4 Viterbi word alignments for the parallel sentences and need to retrain the system, for example to optimize the training parameters, we can include the Viterbi word alignments trained on the original corpora into the binary segmentation. In the monotone case, the model is represented as

$$h_4(j,i,1|f_1^J,e_1^I) = \log\left(\frac{N(f_1^j,e_1^i)+N(f_{j+1}^J,e_{i+1}^I)}{N(f_1^J,e_1^I)}\right),$$

where $N(f_1^j,e_1^i)$ denotes the number of the alignment links inside the matrix $(1,1)$ and (j,i). In the non-monotone case the model is formulated in the same way.

3.6 Word Alignment Concatenation

As described in Section 2, our translation is based on phrases, that means for an input sentence we extract all phrases matched in the training corpus and translate with these phrase pairs. Although the aim of segmentation is to split parallel text into translated segment pairs, but the segmentation is still not perfect. During sentence segmentation we might separate a phrase into two segments, so that the whole phrase pair can not be extracted.

To avoid this, we concatenate the word alignments trained with the segmentations of one sentence pair. During the segmentation, the position of each segmentation point in the sentence is memorized. After training the word alignment model with the segmented sentence pairs, the word alignments are concatenated again according to the positions of their segments in the sentences. The original sentence pairs and the concatenated alignments are then used for the phrase extraction.

Table 1: Corpus Statistics: NIST

		Chinese	English
Train	Sentences	8.64 M	
	Running Words	210 M	226 M
	Average Sentence Length	24.4	26.3
	Vocabulary	224 268	359 623
	Singletons	98 842	156 493
Segmentation	Sentences	17.9 M	
	Running Words	210 M	226 M
	Average Sentence Length	11.7	12.6
	Vocabulary	221 517	353 148
	Singletons	97 062	152 965
Segmentation with Additional Data	Sentences	19.5 M	
	Running Words	230 M	248 M
	Added Running Words	8.0%	8.2%
Evaluation	Sentences	878	3 512
	Running Words	24 111	105 516
	Vocabulary	4 095	6 802
	OOVs (Running Words)	8	658

4 Translation Experiments

4.1 Bilingual Sentences Extraction Methods

In this section, we describe the different methods to extract the bilingual sentence pairs from the document aligned corpus.

Given each document pair, we assume that the paragraphs are aligned one to one monotone if both the source and target language documents contain the same number of paragraphs; otherwise the paragraphs are aligned with the Champollion tool.

Starting from the parallel paragraphs we extract the sentences using three methods:

1. Binary segmentation

   The segmentation method described in Section 3 is applied by treating the paragraph pairs as long sentence pairs. We can use the anchor words model described in Section 3.4 to prefer splitting at punctuation marks.

   The lexicon parameters $p(f|e)$ in Equation 2 are estimated as follows: First the sentences are aligned roughly using the dynamic programming algorithm. Training on these aligned sentences, we get the initial lexicon parameters. Then the binary segmentation algorithm is applied to extract the sentences again.

2. Champollion

   After a paragraph is divided into sentences at punctuation marks, the Champollion tool (Ma, 2006) is used, which applies dynamic programming for the sentence alignment.

3. Combination

   The bilingual corpora produced by the binary segmentation and Champollion methods are concatenated and are used in the training of the translation model. Each corpus is assigned a weight. During the training of the word alignment models, the counts of the lexicon entries are linearly interpolated using the corpus weights.

4.2 Translation Tasks

We will present the translation results on two Chinese-English tasks.

1. On the large data track NIST task (NIST, 2005), we will show improvements using the refined binary segmentation method.

Table 2: Corpus Statistics: FBIS

		Segmentation		Champollion	
		Chinese	English	Chinese	English
Train	Sentences	739 899		177 798	
	Running Words	8 588 477	10 111 752	7 659 776	9 801 257
	Average Sentence Length	11.6	13.7	43.1	55.1
	Vocabulary	34 896	56 573	34 377	55 775
	Singletons	4 775	19 283	4 588	19 004
Evaluation	Sentences	878	3 513	878	3 513
	Running Words	24 111	105 516	24 111	105 516
	Vocabulary	4 095	6 802	4 095	6 802
	OOVs (Running Words)	109	2 257	119	2 309

2. On the FBIS corpus, we will compare the different sentence extraction methods described in Section 4.1 with respect to translation performance. We do not apply the extraction methods on the whole NIST corpora, because some corpora provided by the LDC (LDC, 2005) are sentence aligned but not document aligned.

4.3 Corpus Statistics

The training corpora used in NIST task are a set of individual corpora including the FBIS corpus. These corpora are provided by the Linguistic Data Consortium (LDC, 2005), the domains are news articles. The translation experiments are carried out on the NIST 2002 evaluation set.

As shown in Table 1, there are 8.6 million sentence pairs in the original corpora of the NIST task. The average sentence length is about 25. After segmentation, there are twice as many sentence pairs, i.e. 17.9 million, and the average sentence length is around 12. Due to a limitation of GIZA++, sentences consisting of more than one hundred words are filtered out. Segmentation of long sentences circumvents this restriction and allows us include more data. Here we were able to add 8% more Chinese and 8.2% more English running words to the training data. The training time is also reduced.

Table 2 presents statistics of the FBIS data. After the paragraph alignment described in Section 4.1 we have nearly 81 thousand paragraphs, 8.6 million Chinese and 10.1 million English running words. One of the advantages of the binary segmentation is that we do not loose words during the bilingual sentences extraction. However, we produce sentence pairs with very different lengths. Using Champollion we loose 10.8% of the Chinese and 3.1% of the English words.

4.4 Segmentation Parameters

We did not optimize the log-linear model scaling factors for the binary segmentation but used the following fixed values: $\lambda_1 = \lambda_2 = 0.5$ for the IBM-1 models in both directions; $\lambda_3 = 10^8$, if the anchor words model is is used; $\lambda_4 = 30$, if the IBM-4 model is used. The maximum sentence length is 25.

4.5 Evaluation Criteria

We use four different criteria to evaluate the translation results automatically:

- 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, divided by the reference sentence length.

- 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 differ 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,

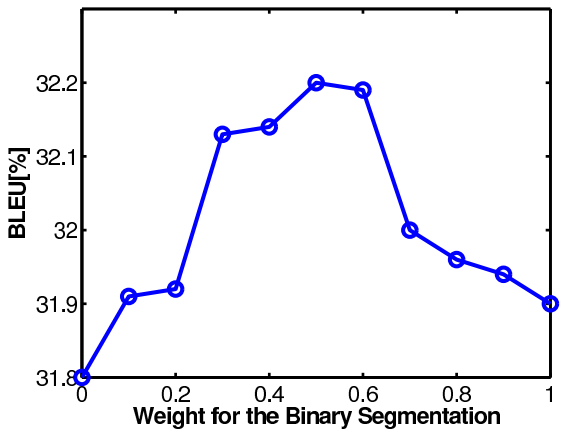

Figure 2: Translation performance as a function of the weight for the binary segmentation α (weight for Champollion: $1 - \alpha$)

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.

4.6 Translation Results

For the segmentation of long sentences into short segments, we performed the experiments on the NIST task. Both in the baseline and the segmentation systems we obtain 4.7 million bilingual phrases during the translation. The method of alignment concatenation increases the number of the extracted bilingual phrase pairs from 4.7 million to 4.9 million, the BLEU score is improved by 0.1%. By including the IBM-4 Viterbi word alignment, the NIST score is improved. The training of the baseline system requires 5.9 days, after the sentence segmentation it requires only 1.5 days. Moreover, the segmentation allows the inclusion of long sentences that are filtered out in the baseline system. Using the added data, the translation performance is enhanced by 0.3% in the BLEU score. Because of the long translation period, the translation parameters are only optimized on the baseline system with respect to the BLEU score, we could expect a further improvement if the parameters were also optimized on the segmentation system.

Our major objective here is to introduce another approach to parallel sentence extraction: binary segmentation of the bilingual texts recursively. We use the paragraph-aligned corpus as a starting point. Table 4 presents the translation results on the training corpora generated by the different methods described in Section 4.1. The translation parameters are optimized with the respect to the BLEU score. We observe that the binary segmentation methods are comparable to Champollion and the segmentation with anchors outperforms the one without anchors. By combining the methods of Champollion and the binary segmentation with anchors, the BLEU score is improved by 0.4% absolutely.

We optimized the weightings for the binary segmentation method, the sum of the weightings for both methods is one. As shown in Figure 2, using one of the methods alone does not produce the best result. The maximum BLEU score is attained when both methods are combined with equal weightings.

5 Discussion and Future Work

We successfully applied the binary sentence segmentation method to extract bilingual sentence pairs from the document aligned texts. The experiments on the FBIS data show an enhancement of 0.4% of the BLEU score compared to the score obtained using a state-of-art sentence aligner. In addition to the encouraging results obtained, further improvements could be achieved in the following ways:

1. By extracting bilingual paragraphs from the documents, we lost running words using Champollion. Applying the segmentation approach to paragraph alignment might avoid the loss of this data.

2. We combined a number of different models in the binary segmentation, such as IBM-1, and anchor words. The model weightings could be optimized with respect to translation quality.

Table 3: Translation Results using Refined Segmentation Methods on NIST task

	Error Rate[%]		Accuracy	
	WER	PER	NIST	BLEU[%]
Baseline	62.7	42.1	8.95	33.5
Segmentation	62.6	42.4	8.80	33.5
Segmentation + concatenation	62.4	42.3	8.84	33.6
Segmentation + concatenation + IBM-4	62.8	42.4	8.91	33.6
Segmentation + added data	62.9	42.5	9.00	33.9

Table 4: Translation Results on Sentence Alignment Task with FBIS Training Corpus

	Error Rate[%]		Accuracy	
	WER	PER	NIST	BLEU[%]
Champollion	64.2	43.7	8.61	31.8
Segmentation without Anchors	64.3	44.4	8.57	31.8
Segmentation with Anchors	64.0	43.9	8.58	31.9
Champollion + Segmentation with Anchors	64.3	44.2	8.57	32.2

3. In the binary segmentation method, an incorrect segmentation results in further mistakes in the segmentation decisions of all its subsegments. An alternative method (Wu, 1997) makes decisions at the end but has a high computational requirement. A restricted expansion of the search space might better balance segmentation accuracy and the efficiency.

6 Acknowledgments

This work was supported by 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) and the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023.

References

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.

Y. Deng, S. Kumar, and W. Byrne. 2006. Segmentation and alignment of parallel text for statistical machine translation. *Natural Language Engineering*, Accepted. To appear.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology*, pages 128–132, San Diego, California, March.

W. A. Gale and K. W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–90.

LDC. 2005. Linguistic data consortium resource home page. http://www.ldc.upenn.edu/Projects/TIDES.

X. Ma. 2006. Champollion: A robust parallel text sentence aligner. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC)*, Genoa, Italy, Accepted. To appear.

NIST. 2005. 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.

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

M. Simard and P. Langlais. 2003. Statistical translation alignment with compositionality constraints. In *NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton, Canada, May.

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

J. Xu, R. Zens, and H. Ney. 2005. Sentence segmentation using IBM word alignment model 1. In *Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation)*, pages 280–287, Budapest, Hungary, May.

R. Zens, O. Bender, S. Hasan, S. Khadivi, E. Matusov, J. Xu, Y. Zhang, and H. Ney. 2005. The RWTH phrase-based statistical machine translation system. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 155–162, Pittsburgh, PA, October.