Natural Language Engineering **12** (4): 1–26. © 2006 Cambridge University Press
doi:10.1017/S1351324906004293 Printed in the United Kingdom

Segmentation and alignment of parallel text for statistical machine translation

YONGGANG DENG
Center for Language and Speech Processing, Department of Electrical and Computer Engineering, The Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218, USA
e-mail: dengyg@jhu.edu

SHANKAR KUMAR
Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
e-mail: shankar.kumar@gmail.com

WILLIAM BYRNE
Department of Engineering, Cambridge University, Trumpington Street, Cambridge CB2 1PZ, UK
e-mail: wjb31@cam.ac.uk

(*Received 14 September 2005; revised 23 December 2005*)

Abstract

We address the problem of extracting bilingual chunk pairs from parallel text to create training sets for statistical machine translation. We formulate the problem in terms of a stochastic generative process over text translation pairs, and derive two different alignment procedures based on the underlying alignment model. The first procedure is a now-standard dynamic programming alignment model which we use to generate an initial coarse alignment of the parallel text. The second procedure is a divisive clustering parallel text alignment procedure which we use to refine the first-pass alignments. This latter procedure is novel in that it permits the segmentation of the parallel text into sub-sentence units which are allowed to be reordered to improve the chunk alignment. The quality of chunk pairs are measured by the performance of machine translation systems trained from them. We show practical benefits of divisive clustering as well as how system performance can be improved by exploiting portions of the parallel text that otherwise would have to be discarded. We also show that chunk alignment as a first step in word alignment can significantly reduce word alignment error rate.

1 Introduction

Parallel texts play an important role in the development of statistical Machine Translation (MT) systems (Brown, Pietra, Pietra and Mercer 1993). A typical training scenario for a translation system starts with a collection of paired sentence translations in the languages of interest. Model-based estimation techniques extract information from this parallel text that is crucial for translation, such as translation lexicons, word-to-word alignments, and phrase translation pairs. Although it is a

crucial first step in such training procedures, sentence alignment is often considered as a separate modeling problem, along with other practical concerns such as text normalization.

This paper discusses a modeling approach which is a first step towards a complete statistical translation model incorporating the alignment of parallel texts. Our goal is to present a general statistical model of large scale chunk alignment within parallel texts, and to develop an iterative language independent chunking procedure when no linguistic knowledge is available. We evaluate our model in context of statistical machine translation.

Extracting chunk pairs is an alignment problem that falls somewhere between word alignment and sentence alignment. It incorporates and extends well-established techniques for sentence alignment with the aim of aligning text at the sub-sentence level. There are two practical benefits to be had from doing this. Shorter segments can lead to quicker training of MT systems since MT training tends to run faster on shorter sentences. Additionally, the ability to break very long sentences into smaller segments will make it possible to train with text that would otherwise have to be discarded prior to training. While these may seem like mainly practical concerns, fast training and thorough exploitation of all available parallel text are crucial for effective development of statistical NLP systems. Beyond these practical concerns, we also provide evidence that word alignments over chunks aligned at the subsentence level can be better than word alignment over sentence pairs.

Many approaches have been proposed to align sentence pairs in parallel text. One widely used method is a dynamic programming procedure based on sentence length statistics (Brown *et al.* 1991; Gale and Church 1991). This approach was extended and improved upon by Chen (Chen 1993) who introduced methods that bootstrap from sentences aligned by hand and incorporate word translation probabilities. Wu (Wu 1994) also extended the length-based method proposed by Gale and Church (1991) to non-Indo-European languages by taking advantage of pre-defined domain specific word correspondences. To reduce reliability on prior knowledge about languages and improve robustness to different domains, Haruno and Yamazaki (1996) iteratively acquired word correspondences during the alignment process with the help of a general bilingual dictionary, and Melamed (1997) developed a geometric approach to alignment based on word correspondences. Typically, there are two desirable properties of sentence alignment: the alignment procedure should be robust to variable quality translations present in the parallel text, and the resulting alignment should be accurate. While accuracy is usually achieved by incorporating lexical cues, robustness can be addressed by bootstrapping with multi-pass search (Simard and Plamondon 1998; Moore 2002), where those sentence pairs with "high quality" are identified and used to refine the models which are then applied to discover more pairs in the whole corpora. There are of course many applications in NLP that rely on aligned parallel text, including statistical bilingual parsing, translation lexicon induction, cross lingual information retrieval (Oard 1996) and language modeling (Kim and Khudanpur 2003).

In this work, we develop a generative model of chunk alignment that can be used to extract and align chunks in parallel text. Within this framework two

alignment algorithms are derived in a straightforward manner. One is a dynamic programming procedure similar to those mentioned above. The second algorithm is a divisive clustering approach to parallel text alignment that begins by finding coarse alignments that are then iteratively refined by successive binary splitting. Both of these algorithms are derived as maximum likelihood search procedures that arise due to variations in the formulation of the underlying model. These are 'source-channel' models in which the source language generates the target language sentences through a variety of stochastic transformations; alternative approaches to simultaneous generation are also possible (Marcu and Wong 2002). We note that this is certainly not the first application of binary search in translation; binary descriptions of sentence structure for translation were explored by Wu (1995, 1997). However, our approach is intended to be much simpler and is developed for very different purposes, namely to prepare parallel text for use in model-based SMT parameter estimation. Our interest here is not in the simultaneous parsing of the two languages. We are simply interested in a general procedure that relies on raw lexical statistics rather than a complex grammar. We sacrifice descriptive power to gain simplicity and the ability to align large amounts of parallel text. We note finally that sentence alignment is a special case of chunk alignment where each chunk consists of single or multiple sentences. Our approach is to develop an end-to-end probabilistic framework for chunk alignment that retains the benefits of more practical engineering approaches while being built on a carefully formulated series of generative models.

We will show that we can obtain improved translation performance through the aggressive use of available parallel text. This work was motivated originally by the development of translation systems submitted to the NIST 2004 and 2005 and TC-STAR 2005 Machine Translation evaluations (NIST 2004; NIST 2005; TC-STAR 2005). The value of making the best use of the available parallel text is widely recognized, and similar approaches to parallel text refinement were subsequently adopted by other researchers (e.g. Xu, Zens and Ney 2005).

In the following sections we will introduce the model and derive the two alignment procedures. We will discuss their application to parallel text alignment and measure their performance by their direct influence on MT evaluation performance as well as through indirect studies of the quality of induced translation lexicons and word alignment error rates of the trained MT systems.

2 A generative model of parallel text segmentation and alignment

We begin with some definitions. A document is divided into a sequence of *segments* which are delimited by *boundary markers* identified within the text. The definition of the boundary markers will vary depending on the alignment task. Coarse segmentation results when boundary markers are defined at sentence or possibly even paragraph boundaries. Finer segmentation results from considering punctuation marks, or even white space, as boundary points, in which case it is possible for a document to be divided into sub-sentential segments. However, once boundary

markers are identified within a document, the segments are specified and are indivisible.

A *chunk* consists of one or more successive segments. Depending on how the segments are defined by the boundary markers, chunks can be formed of multiple sentences, single sentences, phrases, or words; the concept is generic. It is these chunks that are to be aligned as possible translations. If two chunks from the parallel text are hypothesized as possible translations, they are considered to be a *chunk pair*. In this way, document alignment is performed by a deterministic segmentation of the documents, followed by a joint chunking and alignment of the segments.

We emphasize that while this modeling framework is general enough for single word chunks to be aligned, we do not generate alignments at that level of refinement. The model does not include the complex translation and alignment distributions (e.g. distortion and fertility) that would be needed to assure high quality word alignments. Including these components would complicate the alignment and estimation algorithms to the point that the resulting system would be too slow for its intended use, namely the unsupervised coarse alignment of large amounts of parallel text. We note also that these model components which are needed for detailed alignment are not needed for the coarser alignments we intend to produce. We now describe the random variables and the underlying distributions involved in this alignment model.

Alignment Variables The parallel text to be chunk aligned has n segments in the target language (say, Chinese) $\mathbf{t} = \mathbf{t}_1^n$ and m segments in the source language (say, English) $\mathbf{s} = \mathbf{s}_1^m$. Note that each $\mathbf{t}_j$ and $\mathbf{s}_i$ is a segment, which is a string that cannot be further broken down by the aligner.

To describe an alignment between the documents $\mathbf{t}$ and $\mathbf{s}$, we introduce a (hidden) chunk alignment variable $a_1^K(m,n)$ which specifies the alignment of the chunks within the documents. The alignment process is defined and constrained as follows:

A1 The parallel text has m segments in source string $\mathbf{s}$ and n segments in target string $\mathbf{t}$. We use a_1^K as a shorthand for $a_1^K(m,n)$ since we are considering known documents with m and n segments on source and target sides, respectively.

A2 $(\mathbf{s},\mathbf{t})$ is divided into K chunk pairs; K is a random variable.

A3 For each $k = 1,2,\ldots,K$, a_k is a 4-tuple $a_k = (a[k].ss, a[k].se, a[k].ts, a[k].te)$. $a[k].ss$ identifies the starting segment index on the source side, and $a[k].se$ identifies the final index on the the source side; $a[k].ts$ and $a[k].te$ play the same role on the target side. For convenience we introduce $a[k].slen = a[k].se - a[k].ss + 1$ and $a[k].tlen = a[k].te - a[k].ts + 1$ which define the number of segments on each side of the chunk pair.

A4 There are boundary constraints on the chunk alignments:
$a[1].ss = a[1].ts = 1, a[K].se = m, a[K].te = n$

A5 There are continuity constraints on the chunk alignments:
$a[k].ss = a[k-1].se + 1, a[k].ts = a[k-1].te + 1, \quad k = 2,\ldots,K.$

We note that the above conditions require that chunks be paired sequentially; this will be relaxed later to obtain monotonic and non-monotonic versions of the chunk alignment procedure. Consequently, once $\mathbf{s}_1^m$ and $\mathbf{t}_1^n$ are each divided into K chunks, the alignment between them is fixed. Under a given alignment and segmentation, $\mathbf{s}_{a_k}$ and $\mathbf{t}_{a_k}$ denote the k^{th} chunks in the source and target texts, respectively, i.e. $\mathbf{s}_1^m = \mathbf{s}_{a_1} \ldots \mathbf{s}_{a_K}$ and $\mathbf{t}_1^n = \mathbf{t}_{a_1} \ldots \mathbf{t}_{a_K}$.

Generative Chunk Alignment Model The conditional probability of generating $\mathbf{t}$ given $\mathbf{s}$ is

$$P(\mathbf{t}_1^n|\mathbf{s}_1^m) = \sum_{K,a_1^K} P(\mathbf{t}_1^n, K, a_1^K|\mathbf{s}_1^m) \tag{1}$$

and by Bayes Rule we have

$$P(\mathbf{t}_1^n, K, a_1^K|\mathbf{s}_1^m) = P(n|\mathbf{s}_1^m)P(K|\mathbf{s}_1^m, n)P(a_1^K|\mathbf{s}_1^m, n, K)P(\mathbf{t}_1^n|\mathbf{s}_1^m, n, K, a_1^K)\,. \tag{2}$$

This defines the component distributions of the alignment model, as well as their underlying dependencies. We explain these component models in detail, pointing out the simplifying assumptions involved in each.

Source Segmentation Model $P(n|\mathbf{s}_1^m)$ is the probability that the source string generates a target language document with n segments. This is a component distribution, but it is not needed for alignment since the translation segments are determined by the boundary markers within the text to be aligned.

Chunk Count Model $P(K|\mathbf{s}_1^m, n)$ is the probability that there are K chunks when $\mathbf{s}_1^m$ is paired with n segments of the target string. We ignore the words of the string $\mathbf{s}_1^m$ and assume K depends only on m and n: $P(K|\mathbf{s}_1^m, n) \equiv \beta(K|m, n)$.

Chunk Alignment Sequence Model In the alignment process distribution $P(a_1^K|\mathbf{s}_1^m, n, K)$, we make two assumptions:

(a) the chunk alignment a_1^K is independent of the source words, and
(b) the chunk pairs are independent of each other, i.e., each target segment depends only on the source segment to which it is aligned

With these assumptions we have $P(a_1^K|\mathbf{s}_1^m, n, K) = \frac{1}{Z_{m,n,K}} \prod_{k=1}^K p(a_k)$ with the normalization constant $Z_{m,n,K} = \sum_{a_1^K} \prod_{k=1}^K p(a_k)$.

There are many possibilities in defining the alignment distribution $p(a_k)$. One form that we study specifies the range of segment lengths that will be allowed in chunk alignment. If $x = a[k].slen$ and $y = a[k].tlen$, then

$$p(a_k) = p(x, y) = \begin{cases} \frac{1}{g_{\lambda,\alpha}} e^{-\lambda(\alpha(x+y)+(1-\alpha)|x-y|)} & 1 \leq x, y \leq R \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\lambda \geq 0$ and $0 \leq \alpha \leq 1$; R specifies the maximum number of segments that can be incorporated into a chunk. In previous work, this distribution over lengths has been tabulated explicitly (e.g. Wu (1994, Table 1)); we use a parameterized form

mainly for convenience. Setting $\alpha = 0$ favors chunk alignments of equal segment lengths, while $\alpha = 1$ prefers shorter length segments. Setting $\lambda = 0$ specifies a uniform distribution over the allowed lengths.

Target Sequence Model $P(\mathbf{t}_1^n|\mathbf{s}_1^m, n, K, a_1^K)$ is the probability of generating the target string given the source string and the chunk alignment. We derive it from a word translation model with an independence assumption similar to assumption (**b**) of the Chunk Alignment Sequence Model:

$$P(\mathbf{t}_1^n|\mathbf{s}_1^m, n, K, a_1^K) = \prod_{k=1}^{K} P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k}). \tag{4}$$

The chunk-to-chunk translation probability is derived from a simple word translation model. With e_1^v and f_1^u denoting the word sequences for the chunk $\mathbf{s}_{a_k}$ and $\mathbf{t}_{a_k}$, we use IBM Model-1 (Brown *et al.* 1993) translation probabilities to assign likelihood to the translated segments

$$P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k}) = P(f_1^u|e_1^v, u)\, P(u|e_1^v) = \frac{P(u|v)}{(v+1)^u} \prod_{j=1}^{u} \sum_{i=0}^{v} t(f_j|e_i)\ . \tag{5}$$

$t(f_j|e_i)$ is the probability of source word e_i being translated into target word f_j; e_0 is a NULL word. Other formulations are of course possible. However Model-1 treats translations as unordered documents within which any word in the target string can be generated as a translation of any source word, and this is consistent with the lack of structure within our model below the segment level. Model-1 likelihoods are easily computed and the Expectation-Maximization (EM) algorithm can be used to estimate those translation probabilities from a collection of sentence pairs (Brown *et al.* 1993).

The remaining component distribution $P(u|v)$ is the probability that the v words in source string generates a target string of u words. We follow Gale and Church's (1991) model and assume $u - cv$ is normally distributed as

$$\frac{u - cv}{\sqrt{v\sigma^2}} \sim \mathcal{N}(0, 1) \tag{6}$$

where the scalar c is the global length ratio between target language and source language. $P(u|v)$ can be calculated by integrating a standard normal distribution accordingly.

Summary In the preceding presentation we have presented a statistical generative chunk alignment models based on word-to-word translation model. The process to generate a target document $\mathbf{t}_1^n$ from $\mathbf{s}_1^m$ proceeds along the following steps:

1. Choose the number of source language segments n according to probability distribution $P(n|\mathbf{s}_1^m)$.
2. Choose the number of chunk pairs K according to probability distribution $\beta(K|m, n)$.
3. Choose chunk alignment a_1^K according to probability distribution $P(a_1^K|m, n, K)$

4. For each $k = 1, 2, \ldots, K$, produce $\mathbf{t}_{a_k}$ from $\mathbf{s}_{a_k}$ via the word-to-word translation probability $P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k})$.

The above steps are of course only conceptual. In alignment, we have both source and target documents and we search for the best hidden alignment sequence under the model.

3 Chunk alignment search algorithms

We now address the problem of parallel text alignment under the model presented in the previous section. We assume we have parallel text aligned at the document level. We define a set of boundary markers and these uniquely segment the documents into segment sequences $(\mathbf{s}_1^m, \mathbf{t}_1^n)$. The goal is to find the optimal alignment of these segments under the model-based Maximum A Posteriori (MAP) criterion

$$\{\hat{K}, \hat{a}_1^{\hat{K}}\} = \underset{K, a_1^K}{\operatorname{argmax}} P(K, a_1^K | \mathbf{s}_1^m, \mathbf{t}_1^n) = \underset{K, a_1^K}{\operatorname{argmax}} P(\mathbf{t}_1^n, K, a_1^K | \mathbf{s}_1^m). \tag{7}$$

We consider two different alignment strategies. The first is an instance of the widely studied family of dynamic programming sentence alignment procedures (Brown *et al.* 1991; Gale and Church 1991; Wu 1994; Simard and Plamondon 1998; Moore 2002). The second approach is a novel approach to parallel text alignment by divisive clustering. We will show how these very different alignment procedures can be both derived as MAP search procedures under the generative model. Their differences arise from changes in the form of the component distributions within the generative model.

3.1 Monotonic alignment of translated segments via dynamic programming

Sentence alignment can be made computationally feasible through the imposition of alignment constraints. For instance, by insisting that **(a)** a segment in one language align to only 0, 1, or 2 segments in the other language, and that **(b)** the alignment be monotonic and continuous, an efficient dynamic programming alignment algorithm can be found (Gale and Church 1991).

We describe assumptions concerning model introduced in the previous section that make it possible to obtain an efficient dynamic program algorithm to realize the MAP alignment. We note first that obtaining an efficient and optimal chunk alignment procedure $\hat{a}_1^{\hat{K}}$ is not straightforward in general due to the chunk count distribution $\beta(\cdot)$ and the normalization terms $Z_{m,n,K}$. A straightforward implementation would find out optimal alignment for each K, and choose the best one among them. This would require an exhaustive search of exponential complexity over all valid chunk alignments. We describe a particular model formulation under which MAP alignment by dynamic programming is possible and this exponential complexity is avoided.

3.1.1 Simplifying assumptions for efficient monotonic alignment

We assume that the chunk count likelihood $\beta(K|m,n)$ is proportional to the probability of finding an alignment with K chunks, i.e. that it is proportional to the normalization term $Z_{m,n,K}$. It follows, therefore, that

$$\beta(K|m,n) = \frac{Z_{m,n,K}}{Z_{m,n}} \tag{8}$$

where $Z_{m,n} = \sum_{K=1}^{\min(m,n)} Z_{m,n,K}$. Equation 2 then simplifies to

$$P(\mathbf{t}_1^n, K, a_1^K|\mathbf{s}_1^m) = \frac{P(n|\mathbf{s}_1^m)}{Z_{m,n}} \prod_{k=1}^{K} p(a_k) P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k}) \tag{9}$$

and the best chunk alignment is defined as

$$\{\hat{K}, \hat{a}_1^{\hat{K}}\} = \underset{K,a_1^K}{\operatorname{argmax}} \prod_{k=1}^{K} p(a_k) P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k}) . \tag{10}$$

This alignment can be obtained via dynamic programming. Given two chunk prefix sequences $\mathbf{s}_1^i$ and $\mathbf{t}_1^j$, the likelihood of their best alignment is

$$\alpha(i,j) = \max_{k,a_1^k(i,j)} \prod_{k'=1}^{k} p(a_{k'}) P(\mathbf{t}_{a_{k'}}|\mathbf{s}_{a_{k'}}) . \tag{11}$$

which can be computed recursively

$$\alpha(i,j) = \max_{1 \le x,y \le R} \alpha(i-x, j-y) \cdot p(x,y) \cdot P(\mathbf{t}_{j-y+1}^{j}|\mathbf{s}_{i-x+1}^{i}) \tag{12}$$

The dynamic programming procedure searches through an $m \times n$ grid. The search is initialized with $\alpha(0,0) = 1$, and by backtracing from the final grid point (m,n) the optimal chunk alignment can be obtained. In the search, the optimum values of x and y are retained at each (i,j) along with the maximum α value.

We note that if we assume a flat translation table within IBM Model-1, i.e. that any word in the source language segment can be translated with equal probability as any word in the target language segment, then the algorithm is equivalent to dynamic programming based on sentence length (Brown *et al.* 1991; Gale and Church 1991).

3.2 Divisive clustering of translated segments

The previous section describes a particular form of the chunk count distribution that leads to an efficient monotonic alignment. We now describe the alignment procedure that results when this distribution is defined so as to allow only binary segmentation of the documents, i.e. if

$$\beta(K|m,n) = \begin{cases} 1 & K = 2 \\ 0 & \text{otherwise} \end{cases} . \tag{13}$$

Under this distribution the segment sequences that make up each document are grouped into two chunks (two source chunks and two target chunks) which are then

aligned. With the range of K restricted in this way, the chunk alignment sequence contains only two terms: $a_1 a_2$. Given a parallel text $(\mathbf{s}, \mathbf{t})$, a_1^2 will split $\mathbf{s}$ into two chunks $\mathbf{s}_1\mathbf{s}_2$ and $\mathbf{t}$ into two chunks $\mathbf{t}_1\mathbf{t}_2$. Under the model-based MAP criterion, the best split is found by

$$\hat{a}_1^2 = \underset{a_1,a_2}{\operatorname{argmax}}\, p(a_1)p(a_2)P(\mathbf{t}_{a_1}|\mathbf{s}_{a_1})P(\mathbf{t}_{a_2}|\mathbf{s}_{a_2}). \tag{14}$$

For simplicity, $p(a_1)$ and $p(a_2)$ are taken as uniform, although other distributions could be used. The search procedure is straightforward: all possible $m \times n$ binary alignments are considered. Given the simple form of Model-1, statistics can be precomputed so that Equation 5 can be efficiently found over all these pairings.

3.2.1 Iterative binary search and non-monotonic search

The above procedure is optimum for binary splitting and alignment. Of course, with lengthy documents a simple binary splitting and alignment is of limited value. We therefore perform iterative binary splitting in which the document pairs are aligned through a succession of binary splits and alignments: at each step, each previously aligned chunk pair is considered as a 'document' which is divided and split by the above criteria. The idea is to find the best split of each aligned pair into two smaller aligned chunks and then further split the derived chunk pairs as needed. As an alternative to dynamic programming alignment for chunking parallel text, this divisive clustering approach is a divide-and-conquer technique. Each individual splitting is optimal, under the above criteria, but the overall alignment is of course suboptimal, since any binary splitting and pairing performed early on may prevent a subsequent operation which would be preferable. This approach is similar to other divisive clustering schemes, such as can be used in creating Vector Quanitization codebooks (Linde, Buzo and Gray 1980) or decision trees (Breiman, Friedman, Olshen and Stone 1984), in which optimality is sacrificed in favor of efficiency.

The computational simplicity of this style of divisive clustering makes it feasible to consider non-monotonic search. In considering where to split documents, we allow the order of the resulting two chunks to be reversed. This requires a relaxation of the continuity constraint A5 given in section 2, and it does increase search complexity, however we find it useful to incorporate it within a hybrid alignment approach, described next.

3.3 A hybrid alignment procedure

The Dynamic Programming and the Divisive Clustering algorithms arise from very different formulations of the underlying generative model. By appropriately defining the chunk count component distribution, we obtain either of the two procedures. As a result, even though they search over the same potential chunk alignment hypotheses, the algorithms proceed differently and can be expected to yield different answers.

The two algorithms are complementary in nature. The monotonic, dynamic programming algorithm makes alignment decisions on a global scale so that the

resulting alignment is optimal with respect to the likelihood of the underlying model. The divisive clustering procedure is not globally optimal with respect to the overall likelihood, but it does divide each chunk optimally.

Efficient dynamic programming alignment procedures rely on monotonic alignment. This is not asserted as a rule, of course; however allowing for reordering would greatly complicate the search procedure and increase the size of the alignment trellis. In contrast, the relatively simple search space considered at each iteration of divisive clustering makes it feasible to incorporate simple binary reordering; this is possible because each iteration is done independently of its predecessors.

The analysis of the two algorithms suggests a hybrid alignment approach that takes advantage of the respective strengths of each procedure. We align documents by first applying the dynamic programming procedure to align documents at the sentence level. This is done to produce 'coarse chunks' containing as many as four sentences on either side. We then refine this initial alignment by divisive clustering to produce chunks with subsentential segments delimited by punctuation marks, or even by white spaces defining word boundaries. We refer to this hybrid, two stage procedure as 'DP+DC'.

The rationale underlying the hybrid approach is that reordering is more likely to occur at finer levels of alignment. A loose justification for this is that sentences are relatively unlikely to be moved from the beginning of a source document to the end of its target document, whereas subsentence target segments are relatively more likely to appear within a several sentence neighborhood of their origins.

We now report on experiments investigating the nature and performance of these alignment procedures.

4 Experiments

In this section we report on experiments in word alignment and statistical machine translation intended to investigate the behavior and evaluate the quality of alignment procedures we have proposed.

4.1 Unsupervised sentence alignment

Our initial experiments investigate the quality of automatic sentence alignments produced by different model configurations and alignment strategies. We use a collection of 122 document pairs selected at random from the FBIS Chinese/English parallel corpus (NIST 2003); the Chinese sentences were segmented using the LDC word segmenter (Linguistic Data Consortium 2002). The documents were aligned at the sentence level by bilingual human annotators, resulting in a collection of 2,200 aligned Chinese-English sentence pairs. Of these, 76% consist of sentence-to-sentence pairs, approximately 21% consist of one Chinese sentence aligned to many English sentences, and the remainder are many-to-many alignments.

These human alignments serve as the reference against which the quality of automatically generated alignments is measured. Both alignment precision and alignment recall relative to the human references will be reported, and in these

Table 1. *Parallel text used at each iteration of unsupervised sentence alignment. At iteration 0, the entire 122 document collection of parallel text is used. At iterations 1 through 4 the chunk pairs found at the previous iteration are sorted by likelihood and only those with likelihood above the specified threshold are kept for estimation of the Model-1 translation table*

Iteration	Threshold	Surviving Chunk Count	Total Words (Ch/En)
0	–	–	64K/86K
1	0.8	1278 (56.6%)	34K/44K
2	0.005	1320 (59.2%)	35K/45K
3	0.001	1566 (70.2%)	42K/55K
4	0.001	1623 (72.8%)	44K/58K

only exactly corresponding alignments count as correct. For instance, a many-to-one alignment will not be judged as correct even if it covers a one-to-one reference alignment.

4.1.1 Monotonic sentence alignment using sentence length statistics

We generate initial sentence alignments using monotonic dynamic programming procedure as described in section 3.1. In this, as well as in the other experiments described in this section, the boundary markers are defined so that the chunking and alignment procedures operate at the sentence level. The initial alignment is based on sentence length statistics, i.e. with flat Model-1 word translation tables. The global length ratio c in Equation 6 is set based on document-level statistics: we count the total number of words in the Chinese and English documents and set c to be their ratio. We also set the parameters of the Chunk Alignment Sequence Model (Equation 3) to be $\lambda = 3.0$ and $\alpha = 0.9$; these were found to be generally robust values in experiments not reported here. These parameters, c, λ, and α, are all that is needed to perform sentence alignment under Equation 12. The resulting sentence alignment precision and sentence alignment recall are 81% and 83%, shown as Iteration 0 of Table 1.

4.1.2 Iterative alignment and translation table refinement

We use these initial length-based sentence alignments as the basis for more refined alignments (Yarowsky 1995). Since this alignment procedure is 'self-organizing' and does not make use of any sentence aligned training data, we adopt a strategy that uses the model to produce a reliably aligned subset of the training data. From the aligned pairs we selected those with likelihood higher than 0.8 under Equation 6. Approximately 57% of the initial alignments (44K English words/34K Chinese words) survive this filtering.

With these aligned sentences we can use the EM algorithm to refine the IBM Model-1 translation lexicon. Eight iterations of EM are performed to train

Chinese-to-English distribution $t(\cdot|\cdot)$; this number of iterations was chosen based on the empirical observation that training set likelihood and model parameters varied little after eight EM iterations. With these distributions incorporated into Equation 5, replacing the flat translation table as used in section 4.1.1, monotone sentence alignment performance over the entire corpus increases both precision and recall by approximately 4% relative to the initial length-based sentence alignments.

This forms the basis for an unsupervised sentence alignment procedure that allows us to iteratively refine the translation tables. We relax the inclusion threshold over the likelihood of aligned sentence pairs (Equation 5), which gradually increases the size of the parallel text used to estimate the translation tables.

After each reduction in the threshold, we re-estimate the Model-1 translation table using eight iterations of EM. Table 1 shows the amount of parallel text incorporated at each stage, and the corresponding sentence alignment precision and sentence alignment recall are plotted in Figure 1 marked with line 'E'; for iterations 5 and 6, no filtering is performed and the entire parallel text is used.

4.1.3 Length distributions, divisive clustering, and alignment initialization

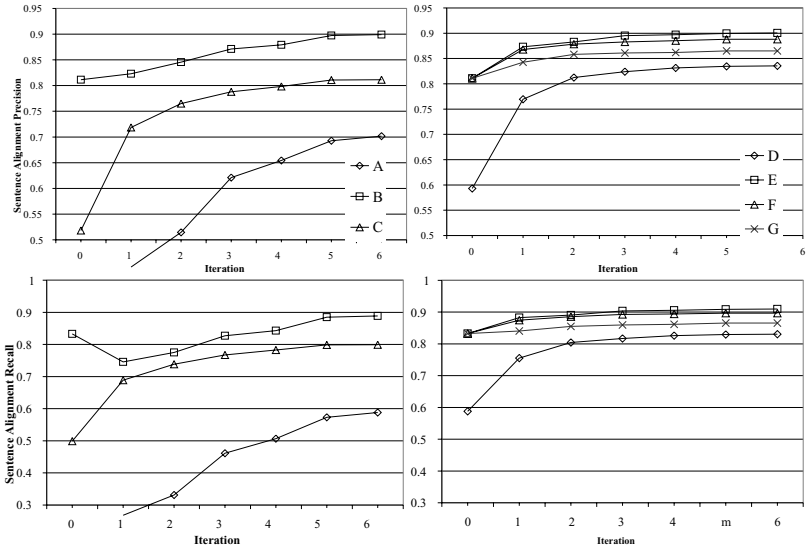
We now investigate the main components of the sentence alignment procedures. These alignment results and search configurations are detailed in Figure 1 in which we present seven search configurations ('A' - 'G') and their alignment performance. Each alignment iteration after iteration 0 involves eight EM iterations, as above, to estimate the Model-1 Chinese-to-English word translation tables; in each scheme, the aligned chunks are filtered at each iteration following the schedule of Table 1.

The procedures are initialized 'naturally': for example, Procedure A is initialized by monotonic sentence alignment based on sentence-length statistics with $\lambda = 0$, and Procedure C is initialized by a single binary split also based on sentence-length statistics. Procedures 'F' and 'G', which incorporate uniform chunk length distributions, are exceptions; they are initialized with the translation table produced by the first iteration of DP+DC($\lambda = 3.0$, $\alpha = 0.9$).

We first note that the DP+DC hybrid search procedure – monotonic sentence alignment to produce coarse alignments which are subsequently refined by division clustering – produces the best overall sentence alignments in terms of sentence alignment precision and recall (see Figure 1, Plot 'E'). Performance is sensitive to the chunk length distribution, and performance suffers if the flat length distribution is used (in Figure 1, Plots 'B' and 'E' show better performance than 'A' and 'D', resp.). Monotone alignment (DP) performs nearly as well under the informative length distribution, although the final alignment recall is slightly worse than that of the DP+DC procedure (in Figure 1, Plot 'E' shows better performance than 'B').

Iterative binary search as an stand-alone alignment algorithm (DC) performs relatively poorly, although it does improve with iterative translation table refinement. Comparison of plots C and G in Figure 1 show that DC alignment is very sensitive to initialization, which is not surprising given the suboptimal nature of its search.

We observe that in nearly all cases the precision and recall increase steadily as the iterations proceed. We also see the value of well-estimated translation tables: when



Plot	Alignment Procedure	Length Distribution Parameters (DP)
A	DP	$\lambda = 0.0$ (uniform)
B	DP	$\lambda = 3.0$, $\alpha = 0.9$
C	DC	n/a
D	DP+DC	$\lambda = 0.0$ (uniform)
E	DP+DC	$\lambda = 3.0$, $\alpha = 0.9$
F	DP+DC	$\lambda = 0.0$ (uniform)
G	DC	n/a

Fig. 1. Precision and recall of automatically sentence alignment procedures over the FBIS sentence-alignment corpus with different initialization and search strategies. Alignment procedures 'F' and 'G' were initialized from iteration 0 of the DP+DC($\lambda = 3.0$, $\alpha = 0.9$) alignment procedure.

initialized with a translation table estimated under slightly stronger models (models with non-uniform chunk-length distributions) the DP+DC and DC procedures both perform better, even when limited by uniform chunk-length distributions (in Figure 1, Plots 'F' and 'G' show better performance than 'D' and 'C', resp.). The implication is that if a reasonable translation lexicon is incorporated in alignment, the chunk length distribution plays less of a role. Loosely speaking, the translation table is more important than the chunk length distributions.

The results of Figure 1 are obtained in an unsupervised manner. No linguistic knowledge is required. This is important and useful in circumstances where linguistic resources such as a bilingual dictionary are not available. Of course, if a bilingual dictionary was available beforehand, it could be used in the initial bootstrapping procedure to provide a translation likelihood to augment the sentence length based alignment likelihood of Equation 6. We note also that the DP+DC alignment procedure achieves a good balance between precision and recall.

4.1.4 Comparable sentence alignment procedures and performance upper bounds

To place these results in context we present the sentence alignment performance obtained on this task by several other well-known algorithms, namely the sentence

Table 2. *Performance of sentence alignment procedures over the FBIS sentence-alignment corpus. Procedures* a, b, c *are unsupervised; Champollion is provided with a Chinese-English translation lexicon; the 'Oracle' version of DP+DC uses Model-1 translation tables trained over the human-aligned sentences*

	Alignment Procedure	Precision	Recall
a	Gale-Church	0.763	0.776
b	Moore'02	0.958	0.441
c	DP+DC($\lambda = 3.0$, $\alpha = 0.9$)	0.901	0.910
d	Champollion	0.937	0.940
e	DP+DC($\lambda = 3.0$, $\alpha = 0.9$) Oracle	0.960	0.971

alignment procedures of Moore (Moore 2002), Gale-Church (Gale and Church 1991), and the Champollion Toolkit (Ma, Cieri and Miller 2004). The results are shown in Table 2. We note that the Moore aligner, as implemented, retains only sentence-to-sentence pairs whose likelihood is above a set threshold; this conservative alignment approach explains its very high alignment precision and lower recall. The Champollion toolkit applies a dynamic programming procedure in alignment. Given a chunk pair, the toolkit looks up a bilingual dictionary and defines the score as the information contained in one side that can be identified by the other side via cross lingual information retrieval (Ma *et al.* 2003). The Champollion aligner requires a bilingual dictionary; we use the 41.8 K entry Chinese-English dictionary distributed with the toolkit. The good performance of Champollion demonstrates the value of a well-crafted bilingual lexicon for the task and languages of interest.

To estimate an upper bound on the performance that might be achieved by sentence alignment procedures based on word-to-word translation, we take the sentence pairs as aligned by humans and use them in estimating the IBM Model-1translation table. We then align the whole collection under this model using one iteration of the DP+DC procedure. This translation table is very heavily biased towards this particular corpus; for instance many translations that would normally appear within the translation table, due for example to different word senses, will not be present unless they happen to occur within this small sample. We therefore call this the 'Oracle' DP+DC condition, and it yields a precision of 96% and a recall of 97%. The upper bound confirms that the IBM Model 1 can indeed be useful for sentence alignment tasks, although this claim must be qualified as above, noting that the translation tables are refined for the task. We emphasize that the 'Oracle DP+DC' has an unfair advantage relative to Champollion in this scenario and no conclusions about their relative power can be drawn from these semi-supervised experiments; the oracle results are included merely as an estimate of the upper bound in alignment performance that might be obtained. It is clear that there is an interdependence between sentence alignment quality and the translation lexicon, and if we use a translation lexicon estimated over human aligned sentences, we can obtain better sentence alignment.

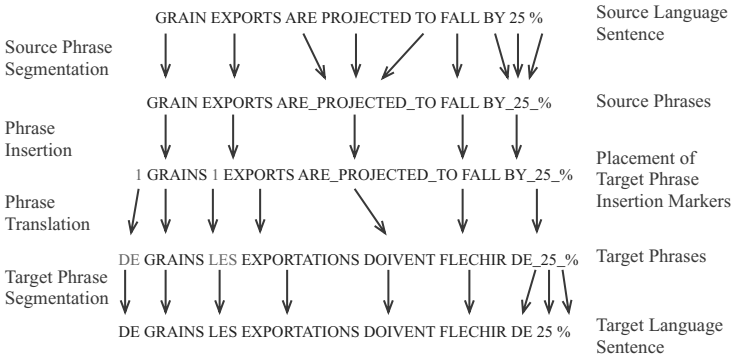


Fig. 2. The translation template model with monotone phrase order. Translation is modeled as a mapping of source language phrase sequences to target language sentences.

4.2 Evaluation via statistical machine translation

Statistical Machine Translation (SMT) systems rely on high quality sentence pairs for training translation models (Brown *et al.* 1993; Och 2002; Kumar *et al.* 2005). We now present experiments to evaluate the influence of chunking alignment algorithms on word alignment and translation performance in SMT. In section 4.2.2 we provide statistics describing the chunk pairs extracted by the various alignment procedures. In the experiments reported in section 4.2.3, we present results showing that divisive clustering does not degrade translation performance; in these controlled experiments, the size of the parallel text and the sentence alignments within are held constant as the alignment procedures vary. In this way we confirm that the divisive clustering procedure can be used to generate parallel text for use in translation model training. Later, in section 4.3, we show that divisive clustering can be used to increase the amount of usable parallel text and that this increase in training data can lead to improved translation. We begin with a brief description of the SMT system used in these experiments.

4.2.1 Translation template model

We use the Translation Template Model (TTM) (Kumar *et al.* 2005) which is a source-channel model of translation with a joint probability distribution over all possible segmentations and alignments of target language sentences and their translations in the source language. The model considers whole phrases rather than words as the basis for translation; translation is in monotone phrase order. The translation process underlying the model is presented in Figure 2. Each of the conditional distributions that make up the model is realized independently and implemented as a weighted finite state acceptor or transducer. Translation of sentences under the TTM can be performed using standard WFST operations involving these transducers.

The Translation Template Model relies on an inventory of target language phrases and their source language translations. These translations need not be unique, in that multiple translations of phrases in either language are allowed. We utilize the

phrase-extract algorithm (Och and Ney 2004) to extract a library of phrase-pairs from word aligned parallel text. We first obtain word alignments of chunk-pairs using IBM Model-4 word level translation models (Brown *et al.* 1993) trained in both translation directions using the Giza++ (Och and Ney 2003), and then form the union of these alignments. We next use the algorithm to identify pairs of phrases in the target and source language that align well according to a set of heuristics (Och and Ney 2004). We will report the word alignment performance of the underlying IBM Model-4 and the translation performance of the TTM system initialized from these models.

4.2.2 Chunking and alignment of parallel text

We present experiments on the NIST Chinese-to-English Translation task (NIST 2003). The goal of this task is the translation of news stories from Chinese to English. The parallel text used for model parameter estimation consists of 11,537 document pairs from the complete FBIS Chinese-English parallel corpus (NIST 2003). The corpus contains 10.8M words in English and 8.03M words in Chinese; the Chinese sentences are segmented into words using the LDC segmenter (Linguistic Data Consortium 2002).

As in the previous section, we investigate the DP+DC hybrid alignment procedure. We align the parallel text by first performing monotonic alignment (DP) under the sentence length model (Equation 6) ($\lambda = 3.0$, $\alpha = 0.9$). In this stage we consider only the end-of-sentence markers as segment boundaries, and insist that each chunk contains at most 4 sentences in either language. From the resulting aligned chunks, we select those chunk pairs with a maximum of 100 words in their English and the Chinese segments; chunks with longer segments are discarded. This yields an aligned text collection of 7.5M Chinese words and 10.1M English words; approximately 10% of the parallel text is discarded. Each aligned chunk pair contains 28 Chinese words and 38 English words on average; see entry 1 of Table 3. We next apply divisive clustering to the chunk pairs obtained by DP. In this step, we consider all punctuations, such as comma, pause, as segment boundary markers. This allows for a much finer alignment of sentence segments (Table 3, entry 2).

Using the chunk pairs produced by length based model with divisive clustering (Table 3, entry 2), we train IBM Model 1 word translation models. Although it departs from the strict model formulation, we have found it beneficial to training IBM Model 1 translation tables in both translation directions, i.e. from English-to-Chinese and from Chinese-to-English. A single translation table is formed as by finding $\sqrt{P(t|s)P(s|t)}$, and then normalizing appropriately.

We then repeat the DP and DP+DC procedures incorporating these IBM Model-1 translation tables from Step 2; during DP monotone alignment, we set the parameter $\lambda = 0$ in Equation (3) to allow chunk pairs to align freely.

We observe that the training text in System 2 is derived from that of System 1 by divisive clustering. System 2 retains all the text aligned by System 1, but produces pairs of shorter chunks. A similar relationship holds between Systems 3 and 4. We will use the aligned text collections produced by these strategies in training

Table 3. *Aligned chunk pair statistics over contrastive alignment configurations. Step 1: initial chunk alignments obtained by DP monotone alignment using sentence length statistics. Step 2: divisive clustering of aligned chunks from Step 1 under sentence-length statistics. The aligned chunks at Step 2 are used in training a Model-1 translation table; this table is held fixed for Steps 3 and 4. Step 3: chunk alignments obtained by DP monotone alignment using Model-1 translation table. Step 4: divisive clustering of aligned chunks from Step 1 under Model-1 translation table*

	Alignment Procedure	Chunk Translation Model	Words (M) Ch/En	Average words per chunk Ch/EN	GIZA++ Model-4 Training Time (CPU hrs)
1	DP	length-based	7.5/10.1	28/38	20
2	DP+DC	length-based	7.5/10.1	20/27	9
3	DP	Model 1	7.2/9.7	29/40	21
4	DP+DC	Model 1	7.2/9.7	16/22	8

SMT systems that will be used for word-level alignment and phrase-based statistical machine translation.

4.2.3 Word alignment and translation performance

For each collection of parallel text produced by the four alignment strategies, we use the GIZA++ Toolkit (Och and Ney 2004) to train IBM Model-4 translation models (Brown *et al.* 1993) in both translation directions. The IBM Model-4 training time is also displayed in Table 3. We observe that after applying DC, average chunk size on both sides reduces and this significantly speeds up the MT training procedure. Since the number of distinct word alignments between two sentences is exponential in the sentence length, shorter chunk pairs naturally yield reduced training times. This is an extremely valuable practical benefit of divisive clustering at the subsentence level relative to monotone sentence alignment.

We now measure the word alignment performance of the resulting IBM Model-4 word translation models. Our word alignment test set consists of 124 sentence pairs from the NIST 2001 dry-run MT-eval set (NIST 2003) that are word aligned by a Chinese/English bilingual linguist. Word links are marked only as 'sure', and not as 'probable' as is also sometimes done (Och and Ney 2004). Word alignment performance is measured using the Alignment Error Rate (AER) metric (Och and Ney 2004). For each system described in Table 3, Table 4 shows the AER of IBM Model-4 trained in both translation directions. We observe that the chunk pairs extracted using IBM Model 1 translation tables in parallel text alignment yield lower AER than the sentence length based alignment procedures. We also note that in some cases divisive clustering yields some minor improvement relative to monotonic sentence alignment, and that performance is otherwise comparable.

Table 4. *Word alignment and translation performance corresponding to IBM Model-4 estimated over parallel text collections produced by contrastive alignment configurations. Alignment Error Rates are provided in both translation directions. Translation performance is given as BLEU(%) scores of phrase-based SMT systems based on phrases extracted from the word aligned text*

	Alignment Error Rate (%)		Translation Performance		
Collection	$E \rightarrow C$	$C \rightarrow E$	Eval01	Eval02	Eval03
1	38.6	35.3	25.1	23.1	22.2
2	38.1	35.1	24.7	23.1	22.4
3	38.0	33.6	25.3	23.3	22.3
4	37.1	33.8	25.1	23.5	22.7

We next measure translation performance of a TTM system trained on the four parallel text collections. We report performance on the NIST 2001, 2002 and 2003 evaluation sets that consist of 993, 878, and 919 sentences, respectively. There are four reference translations for each Chinese sentence in these sets, and translation performance is measured using the BLEU metric (Papineni *et al.* 2002).

We use a trigram word language model estimated using modified Kneser-Ney smoothing as implemented in the SRILM toolkit.[1] Our language model training data comes from English news text derived from two sources: online archives (Sept 1998 to Feb 2002) of *The People's Daily*[2] (16.9M words) and the English side of the Xinhua Chinese-English parallel corpus (NIST 2003) (4.3M words). The total language model corpus size is 21M words.

For each of the word-aligned text collections, we show the translation performance of the phrase-based SMT system built on the word alignments (Table 4). We observe that IBM Model-1 yields small improvements over the length based model on all the test sets. Divisive clustering yields comparable performance as sentence level alignment, but with greatly reduced training times. We conclude that the DP+DC procedure has practical benefit relative sentence-length based alignment.

4.3 Maximizing the aligned translations available for training

The controlled experiments in section 4.2 show that applying divisive clustering to derive shorter chunk pairs significantly reduces MT training time while maintaining MT system performance as evaluated by automatic word alignment and translation performance. As a practical matter in Model-4 estimation, very long aligned chunk pairs are simply discarded. The motivation for doing so is to control the memory usage of the training procedure. Additionally, the word alignments generated over

[1] http://www.speech.sri.com/projects/srilm/
[2] http://www.english.people.com.cn

Table 5. *Percentage of usable Arabic-English translations. English tokens for Arabic-English news and UN parallel corpora under different alignment procedures*

Corpus	DP	DP+DC(I)	DP+DC(II)
News	60%	67%	98%
UN	74%	78%	98%

very long segments tend to be unreliable; we will address this last point in section 4.4. In this section, we show another advantage of the two stage alignment and chunking procedure: its ability to align almost all parallel text available for training MT systems. We present experiments on an Arabic-English SMT system to show that these chunk alignment procedures make the most of the available parallel text and that this leads to improved system performance.

The original document pairs consist of news and UN parallel corpora released by LDC (NIST 2003). Before alignment, all Arabic documents are preprocessed by modified Buckwalter tokenizer (Ma *et al.* 2004). There are about 2.95M Arabic tokens and 3.59M English tokens in the News collection; the UN collection consists of 123.16M Arabic tokens and 131.38M English tokens.

We set the maximum number of tokens on both Arabic and English sides to be 60 in GIZA++ model training. If any side has more than 60 tokens or if the chunk pair length ratio is more than 9, the sentence pairs then would have to be discarded; these filtering criteria would discard 45% of the parallel text if no further processing is applied. These criteria enforce practical constraints which prevent the GIZA++ training procedure from running out of memory.

Two iterations of the monotonic DP sentence alignment algorithm (as in Figure 1, plot E) are applied to the Arabic-English document pairs. In the sentence aligned text that results, we find that about 60% and 74% (in terms of English tokens) of all the parallel text can be used in training for the News and UN corpora, respectively (Table 5). This is simply because Arabic sentences tend to be very long. We then apply divisive clustering to these sentence pairs. On the English side, all punctuation marks are considered as boundary markers. On Arabic side, two boundary marker sets are investigated. In one configuration (DP+DC(I)), punctuation serves as boundary markers; in the second configuration (DP+DC(II)), all Arabic tokens are considered as potential boundaries, i.e. white space is also used as boundary markers.

When applying DC with bounary definition DP+DC(I), the statistics of Table 5 show that relatively little of the aligned text is extracted relative to the initial sentence alignment. However, under the more aggressive segmentation scheme of DP+DC(II), almost all avaliable parallel text can be extracted and aligned for use in MT training. It rarely happens that either DP+DC variant requires subsequent filtering of the parallel text to satisfy the length requirements for GIZA++ processing. In the experiments reported here, only 2% of the parallel text is discarded. These instances may well be spurious text, as is inevitably present in large parallel text collections,

Table 6. *Translation performance of TTM Arabic-English systems based on parallel text collections extracted by the alignment procedures*

Corpus	Alignment Procedure	Eval02	Eval03	Eval04
News	As Distributed	30.7	31.9	29.5
	DP	36.3	38.1	34.1
	DP+DC(I)	36.8	38.3	34.2
	DP+DC(II)	37.3	39.3	35.0
News+UN	DP+DC(I)	37.7	39.5	35.5
	DP+DC(II)	38.1	40.1	35.9

and rather than continue with aggressive divisive clustering, we simply discard the small amounts of parallel text that fail to form chunk pairs of the required size.

We also show the advantage of having more parallel text for statistical machine translation system training. The test sets are NIST 2002, 2003 and 2004 Arabic/English MT evaluation sets, which consist of 1043, 663 and 1353 Arabic sentences, respectively. The task is to translate Arabic sentences into English; for each test sentence, there are four English reference translations. As with the Chinese-English MT systems, we perform decoding by the Translation Template Model (TTM). The English language model is a trigram word language model estimated using modified Kneser-Ney smoothing with 400M English words drawn from the English side of the parallel text and the LDC English Gigaword collection.

Phrase translations are extracted from the News and from the UN collections. Performance of the resulting translation systems are shown on each evaluation set in Table 6. Over all test sets, DP+DC(II) performs better than DP+DC(I); this is due to the retention of translations that otherwise would have to be discarded in training. Both techniques perform better than DP alone, and all of these techniques perform much better than simply filtering the distributed translations which, as mentioned, discards 45% of the data. We also note that significant improvements over all test sets are obtained when UN corpora is included in model training.

4.4 Improved subsentence alignment can improve word alignment

Word and sentence alignment are typically addressed as distinct problems to be solved independently, with sentence alignment sometimes even regarded as merely as a text preprocessing step to be done prior to word alignment. The two tasks are of course quite different. As discussed here and in earlier work, sentences in parallel documents can be accurately aligned using algorithms based on relatively simple models, such as IBM Model-1. However, word alignment algorithms require more sophisticated alignment models, such as Hidden Markov Models (Vogel *et al.* 1996) or IBM models with fertility and distortion (Brown *et al.* 1993). An intuitive explanation for this difference is that capturing alignment variation in parallel text

is more challenging as the granularity of the problem becomes smaller. However the interaction between the two types of alignment procedures has not been widely studied.

The experiments reported here investigate the extent to which sub-sentence chunk alignment can improve word alignment. Rather than deriving word alignments directly via Model-4 between manually aligned sentence pairs, we first identify and align chunks at the sub-sentence level and then align the words within the chunks using Model-4.

There is of course a risk in this approach. If chunks are aligned incorrectly, then some correct word alignments are ruled out from the start, since words cannot be aligned across chunk pairs. In this situation, we say that a word alignment is prevented from crossing a chunk alignment boundary. However, if the automatic chunking procedure does a good job both in deciding where and when to split the sentences, then the sub-sentence aligned chunks may actually help guide the word alignment procedure that follows.

Our training text is the complete FBIS Chinese/English parallel corpus, and the test set is the same as that used in the experiments of section 4.2.3. To generate the Model 1 Chinese-to-English translation lexicons needed by the alignment procedures we run GIZA++(Och and Ney 2003) with 10 iterations of EM over the training text. In aligning the test set, boundary points are set at punctuation marks for both the monotone (DP) and divisive clustering (DC) alignment procedures. For the DP procedure, we set $\lambda = 3.0$ and $\alpha = 0.9$ and perform chunk alignment as specified by Equation 10. In DC alignment, we proceed by Equation 14. The recursive parallel binary splitting stops when neither chunk can be split or when one side has less than 10 words and the other side has more than 20 words.

The word alignment performance resulting from these procedures is shown in Table 7. We see first that the divisive clustering procedure generates the shortest subsentence segments of all the procedures. We also see that in all instances except one, DC chunk alignment leads to better quality Model-4 word alignment than the other two procedures, and that both subsentence alignment procedures improve the quality of Model-4 word alignments.

This result suggests the proposed two-stage word alignment strategy can indeed improve word alignment quality relative to the usual word alignment procedure in which word links are established directly from given sentence pairs. To explain where the improvements come from, we inspect the English→Chinese word alignments and analyze the distribution of word links that cross the segment boundaries found by divisive clustering, the most aggressive of the segmentation procedures. The results are presented in Table 8.

First, we note that only ∼2.4% of the reference (manual) word alignment links cross the chunk alignment boundaries found by the divisive clustering procedure. This small fraction further confirms that the DC procedure yields good alignments of sub-sentence chunks: nearly all of the manually generated alignment links between the words in the sentences can be found in these chunk pairs. It follows that an automatic word alignment system is not necessarily handicapped if it aligns only these subsentence chunks and not the original sentence pairs.

Table 7. *Influence of subsentence alignment on alignment error rate*

		Sentence Aligned Test Set	Automatically Aligned Subsentence Chunks	
			DP	DC
Average Ratio of Aligned Segment Lengths (Ch/En words)		24/33	14/19	10/14
	Model-4 Word Alignment Performance			
En→Ch	Precision	67.6	72.2	75.6
	Recall	46.3	48.7	49.6
	AER	45.0	41.8	40.1
Ch→En	Precision	66.5	69.3	72.6
	Recall	59.4	60.4	60.1
	AER	37.3	35.4	34.2

Table 8. *English-to-Chinese word alignment links accuracy relative to chunk alignment boundaries found by divisive clustering*

	Word Alignment Links Relative to DC Chunk Alignment Boundaries	Total Links	Correct Links	Alignment Precision
Manual Word Alignment	Crossing Boundaries	91		
	Within Aligned Chunks	3655		
	All	3746		
DC + Model-4	Within Aligned Chunks	2455	1857	75.6%
Sentence Aligned Test Set+ Model-4	Crossing Boundaries	150	34	22.6%
	Within Aligned Chunks	2415	1701	70.4%
	All	2565	1735	67.6%

We now look at the performance of the Model-4 word alignments relative to the chunk alignment boundaries produced by divisive clustering. Obviously, in the DC case all word alignments generated respect these boundaries, and the precision of 75.6% agrees with the result of Table 7. When applied to sentence pairs, Model-4 is free to generate word alignments that cross the DC chunk alignment boundaries. However, when it does so, errors are likely: there are 150 cross-boundary links, and most of them (77.3%) are incorrect. In fact, if we remove these cross-boundary links, we can improve the alignment precision to 70.4% and reduce the AER to 44.8% from 45.0%.

However, simply applying Model-4 Viterbi word alignment to the subsentence chunks is more effective than discarding links that cross DC chunk alignment boundaries. The result is a great number of correct word alignment links (1857 vs.

1701), higher precision (75.6%) and recall (1857/3746 = 49.6%), and a lower overall AER of 40.1%

These results support the notion that the power of the underlying alignment model should be matched to the alignment task to which it is applied. Model-4 is certainly a better word alignment model that Model-1, yet we still find that chunk alignment procedures based on Model-1 can be used to guide Model-4 word alignment. We take this as evidence that, from the point of view of building statistical models, word and sentence alignment are not independent tasks.

4.5 Translation lexicon induction

As a final experiment to understand the properties of these chunk alignment procedures, we evaluate the quality of the probabilistic translation lexicons they produce. Translation lexicons serve as a bridge between languages and thus play an important role in cross lingual applications. For example, statistical methods of extracting lexicons from parallel (Melamed 2000) and non-parallel corpora (Mann and Yarowsky 2001) have been investigated in the literature, and in section 4.1 we have shown that the quality of sentence alignment improves with the quality of the lexicon used.

We created three subsets of the FBIS Chinese/English translation collection, consisting of 100, 300, 500 document pairs. Over each collection, and over the full FBIS collection, we performed the iterative unsupervised sentence alignment procedure of section 4.1. We then used each collection of aligned text in performing 8 EM iterations to produce an IBM Model 1 Chinese-to-English lexicon.

We measure the precision of these lexicons against the LDC Chinese-to-English dictionary.[3] In doing so, we apply a pruning threshold to the translation probability: if the probability of a translation is below the threshold, it is discarded. In Figure 3, we plot the precision of induced translation lexicon against its size as the pruning threshold varies. The results are consistent with intuition about how these procedures should behave. Overall precision increases with the amount of translated text used in training; as the amount of translations increases, more translations are found at a fixed posterior pruning threshold; and overall precision tracks the posterior level fairly closely. While we observe that it is possible to generate a small, accurate lexicon with 500 document pairs, these experiments do show the limitations of the overall approach: if a translation precision of 0.7 is desired, training with the entire FBIS collection itself still yields fewer than 1000 entries.

5 Conclusion

In this work we have investigated statistical models of parallel text alignment with particular emphasis on the 'chunk count' component distribution. Depending on how this distribution is defined, alignment under a maximum likelihood criteria leads to very different types of alignment search strategies. A chunk count distribution that

[3] LDC Catalog Number LDC2002L27

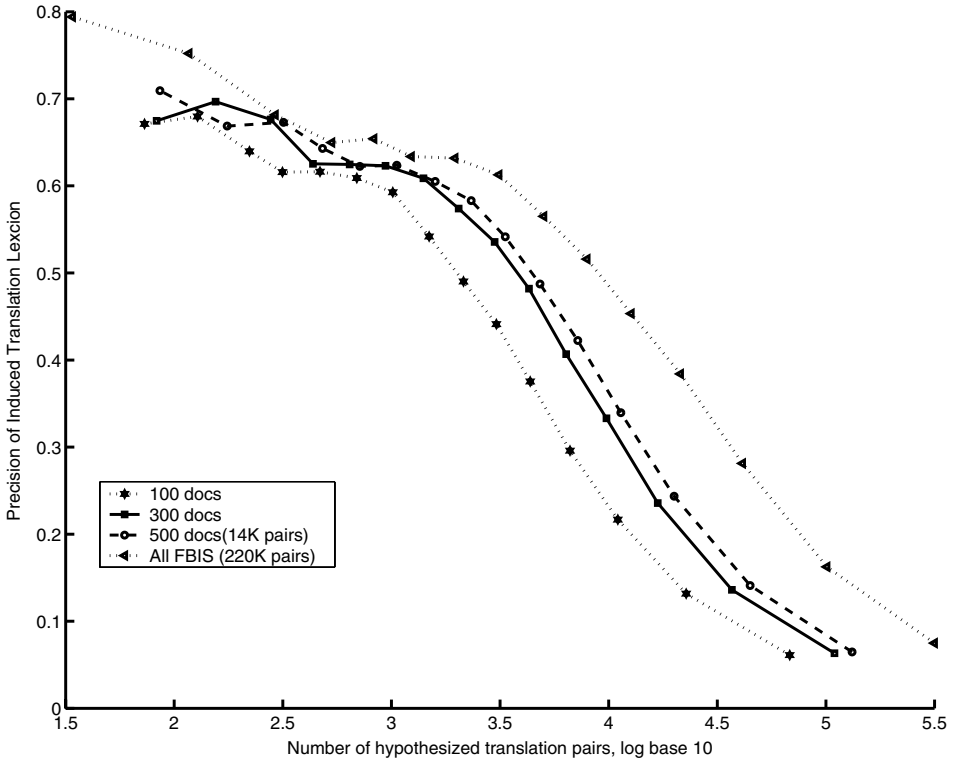


Fig. 3. Precision of induced IBM Model 1 lexicons measured against the LDC Chinese-to-English bilingual dictionary. Each curve is associated with a single alignment of the parallel text. DP+DC algorithm is applied to 100, 300, 500 and all document pairs from FBIS Chinese/English parallel corpus. From each set of alignments eight iterations of EM are used to induce an IBM Model 1 lexicon. Each curve is obtained by pruning the lexicon by a sequence of thresholds on the translation probability. Each point on each curve represents a pruned lexicon. The precision of each of these lexicons is plotted versus its number of entries.

allows a detailed, monotone alignment of sentence segments can lead to dynamic programming alignment procedures of the sort that have been widely studied in previous work. If the distribution is defined so that only binary segmentation and alignment is allowed under the model, we obtain an iterative search procedure. We find that these two types of alignment procedures complement each other and that they can be used together to improve the overall alignment quality.

This hybrid modeling approach was developed with the goal of aligning large parallel text collections for Statistical Machine Translation parameter estimation. We find the approach to be robust in these applications, and when assessed in terms of sentence alignment on a manually annotated test set, we find a balanced performance in precision and recall. An important feature of the approach is the ability to segment at the sub-sentence level as part of the alignment process. We find that this does not degrade translation performance of the resulting systems, even though the sentence segmentation is done with a weak translation model. The

practical benefits of this are faster training of MT systems and the ability to retain more of the available parallel text in MT training.

Beyond the practical benefits of better text processing for SMT parameter estimation, we observed interesting interactions between the word-level and sentence-level alignment procedures we studied. Although the models used in coarse, sentence-level alignment are relatively simple models of translation, they can still guide the alignment of long stretches of text by more powerful translation models based on complicated models of word movement. This suggests that sentence alignment and word alignment are not entirely independent modeling problems, and this work is intended to provide a framework, and the motivation, within which the joint modeling of both problems can be studied.

References

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and regression trees*. Wadsworth International Group.

Brown, P. F., Lai, J. C. and Mercer, R. L. (1991) Aligning Sentences in Parallel Corpora. *Pages 169–176 of: Meeting of the Association for Computational Linguistics.*

Brown, P. F., Pietra, S. A. D., Pietra, V. D. J. and Mercer, R. L. (1993) The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, **19**: 263–312.

Chen, S. F. (1993) Aligning Sentences in Bilingual Corpora using Lexical Information. *Pages 9–16 of: Meeting of the Association for Computational Linguistics.*

Gale, W. A. and Church, K. W. (1991) A Program for Aligning Sentences in Bilingual Corpora. *Pages 177–184 of: Meeting of the Association for Computational Linguistics.*

Haruno, M. and Yamazaki, T. (1996) High-performance bilingual text alignment using statistical and dictionary information. *Pages 131–138 of: Proceedings of ACL '96.*

Kim, W. and Khudanpur, S. (2003) Cross-Lingual Lexical Triggers in Statistical Language Modeling. *Pages 17–24 of: Proc. of EMNLP.*

Kumar, S., Deng, Y. and Byrne, W. (2005) A Weighted Finite State Transducer Translation Template Model for Statistical Machine Translation. *Journal of Natural Language Engineering*, **11**(3).

Linde, Y., Buzo, A. and Gray, R. (1980) An Algorithm for Vector Quantizer Design. *IEEE Transaction on Communications*, **28(1)**, 84–94.

Linguistic Data Consortium. (2002) *LDC Chinese Segmenter*. http://www.ldc.upenn.edu/Projects/Chinese.

Ma, X., *et al.* (2003) Rosetta: a sentence aligner for imperfect world. *Machine Translation Evaluation Workshop*. NIST, Gaithersburg, MD.

Ma, X., Cieri, C. and Miller, D. (2004) Corpora & Tools for Machine Translation. *Machine Translation Evaluation Workshop*. NIST, Alexandria, VA.

Mann, G. and Yarowsky, D. (2001) Multipath Translation Lexicon Induction via Bridge Languages. *Proceedings of the Second Conference of the North American Association for Computational Linguistics.*

Marcu, D. and Wong, W. (2002) A phrase-based, joint probability model for statistical machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002).*

Melamed, I. D. (1997) A Portable Algorithm for Mapping Bitext Correspondence. *Pages 305–312 of: Proceedings of the 35th Annual Conference of the Association for Computational Linguistics.*

Melamed, I. D. (2000) Models of translational equivalence among words. *Computational Linguistics* **26**(2): 221–249.

Moore, R. C. (2002) Fast and Accurate Sentence Alignment of Bilingual Corpora. *Pages 135–244 of: Proceeding of 5th Conference of the Association for Machine Translation in the Americas.*

NIST. (2003) *The NIST Machine Translation Evaluations.* http://www.nist.gov/speech/tests/mt/.

NIST. (2004) *The NIST Machine Translation Evaluations Workshop.* Gaithersburg, MD. http://www.nist.gov/speech/tests/mt/.

NIST. (2005) *The NIST Machine Translation Evaluations Workshop.* North Bethesda, MD. http://www.nist.gov/speech/tests/summaries/2005/mt05.htm.

Oard, D. (1996) *A survey of Multilingual Text Retrieval.* Tech. rept. UMIACS-TR-96-19 CS-TR-3615, University of Maryland, College Park.

Och, F. (2002) *Statistical Machine Translation: From Single Word Models to Alignment Templates.* PhD thesis, RWTH Aachen, Germany.

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

Och, F. J. and Ney, H. (2004) The alignment template approach to statistical machine translation. *Computational Linguistics* **30**(4): 417–449.

Papineni, K., Roukos, S., Ward, T. and Zhu, W. (2002) Bleu: a Method for Automatic Evaluation of Machine Translation. *Pages 311–318 of: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).*

Simard, M. and Plamondon, P. (1998) Bilingual Sentence Alignment: Balancing Robustness and Accuracy. *Pages 59–80 of: Machine Translation*, vol. 13.

TC-STAR. (2005) *TC-STAR Speech-to-Speech Translation Evaluation Meeting.* Trento, Italy. http://www.tc-star.org/.

Vogel, S., Ney, H. and Tillmann, C. (1996) HMM Based Word Alignment in Statistical Translation. *Pages 836–841 of: Proc. of the COLING.*

Wu, D. (1994) Aligning a Parallel English-Chinese Corpus Statistically with Lexical Criteria. *Pages 80–87 of: Meeting of the Association for Computational Linguistics.*

Wu, D. (1995) An Algorithm for Simultaneously Bracketing Parallel Texts by Aligning Words. *Pages 244–251 of: Meeting of the Association for Computational Linguistics.*

Wu, D. (1997) Stochastic inversion transduction grammar and bilingual parsing of parallel corpora. *Computational Linguistics* **23**(3): 377–482.

Xu, J., Zens, R. and Ney, H. (2005) Sentence Segmentation Using IBM Word Alignment Model 1. *Proceedings of the European Association for Machine Translation (EAMT 2005).*

Yarowsky, D. (1995) Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Pages 189–196 of: Meeting of the Association for Computational Linguistics.*