ALIGNING SENTENCES IN BILINGUAL CORPORA USING LEXICAL INFORMATION

Stanley F. Chen
Aiken Computation Laboratory
Division of Applied Sciences
Harvard University
Cambridge, MA 02138
Internet: sfc@calliope.harvard.edu

Abstract

In this paper, we describe a fast algorithm for aligning sentences with their translations in a bilingual corpus. Existing efficient algorithms ignore word identities and only consider sentence length (Brown *et al.*, 1991b; Gale and Church, 1991). Our algorithm constructs a simple statistical word-to-word translation model on the fly during alignment. We find the alignment that maximizes the probability of generating the corpus with this translation model. We have achieved an error rate of approximately 0.4% on Canadian Hansard data, which is a significant improvement over previous results. The algorithm is language independent.

1 Introduction

In this paper, we describe an algorithm for aligning sentences with their translations in a bilingual corpus. Aligned bilingual corpora have proved useful in many tasks, including machine translation (Brown *et al.*, 1990; Sadler, 1989), sense disambiguation (Brown *et al.*, 1991a; Dagan *et al.*, 1991; Gale *et al.*, 1992), and bilingual lexicography (Klavans and Tzoukermann, 1990; Warwick and Russell, 1990).

The task is difficult because sentences frequently do not align one-to-one. Sometimes sentences align many-to-one, and often there are deletions in one of the supposedly parallel corpora of a bilingual corpus. These deletions can be substantial; in the Canadian Hansard corpus, there are many deletions of several thousand sentences and one deletion of over 90,000 sentences.

Previous work includes (Brown *et al.*, 1991b) and (Gale and Church, 1991). In Brown, alignment is based solely on the number of words in each sentence; the actual identities of words are ignored. The general idea is that the closer in length two sentences are, the more likely they align. To perform the search for the best alignment, dynamic programming (Bellman, 1957) is used. Because dynamic programming requires time quadratic in the length of the text aligned, it is not practical to align a large corpus as a single unit. The computation required is drastically reduced if the bilingual corpus can be subdivided into smaller chunks. Brown uses *anchors* to perform this subdivision. An anchor is a piece of text likely to be present at the same location in both of the parallel corpora of a bilingual corpus. Dynamic programming is used to align anchors, and then dynamic programming is used again to align the text between anchors.

The Gale algorithm is similar to the Brown algorithm except that instead of basing alignment on the number of *words* in sentences, alignment is based on the number of *characters* in sentences. Dynamic programming is also used to search for the best alignment. Large corpora are assumed to be already subdivided into smaller chunks.

While these algorithms have achieved remarkably good performance, there is definite room for improvement. These algorithms are not robust with respect to non-literal translations and small deletions; they can easily misalign small passages because they ignore word identities. For example, the type of passage depicted in Figure 1 occurs in the Hansard corpus. With length-based alignment algorithms, these passages may well be misaligned by an even number of sentences if one of the corpora contains a deletion. In addition, with length-based algorithms it is difficult to automatically re-

```
Mr. McInnis?         M. McInnis?
Yes.                 Oui.
Mr. Saunders?        M. Saunders?
No.                  Non.
Mr. Cossitt?         M. Cossitt?
Yes.                 Oui.
   ⋮                    ⋮
```

Figure 1: A Bilingual Corpus Fragment

cover from large deletions. In Brown, anchors are used to deal with this issue, but the selection of anchors requires manual inspection of the corpus to be aligned. Gale does not discuss this issue.

Alignment algorithms that use lexical information offer a potential for higher accuracy. Previous work includes (Kay, 1991) and (Catizone *et al.*, 1989). However, to date lexically-based algorithms have not proved efficient enough to be suitable for large corpora.

In this paper, we describe a fast algorithm for sentence alignment that uses lexical information. The algorithm constructs a simple statistical word-to-word translation model on the fly during sentence alignment. We find the alignment that maximizes the probability of generating the corpus with this translation model. The search strategy used is dynamic programming with thresholding. Because of thresholding, the search is linear in the length of the corpus so that a corpus need not be subdivided into smaller chunks. The search strategy is robust with respect to large deletions; lexical information allows us to confidently identify the beginning and end of deletions.

2 The Alignment Model

2.1 The Alignment Framework

We use an example to introduce our framework for alignment. Consider the bilingual corpus $(\mathcal{E}, \mathcal{F})$ displayed in Figure 2. Assume that we have constructed a model for English-to-French translation, *i.e.*, for all E and F_p we have an estimate for $P(F_p|E)$, the probability that the English sentence E translates to the French passage F_p. Then, we can assign a probability to the English corpus $\mathcal{E}$ translating to the French corpus $\mathcal{F}$ with a particular alignment. For example, consider the alignment $\mathcal{A}_1$ where sentence E_1 corresponds to sentence F_1 and sentence E_2 corresponds to sentences F_2 and F_3. We get

$$P(\mathcal{F}, \mathcal{A}_1|\mathcal{E}) = P(F_1|E_1)P(F_2, F_3|E_2),$$

assuming that successive sentences translate independently of each other. This value should be relatively large, since F_1 is a good translation of E_1 and (F_2, F_3) is a good translation of E_2. Another possible alignment $\mathcal{A}_2$ is one where E_1 maps to nothing and E_2 maps to F_1, F_2, and F_3. We get

$$P(\mathcal{F}, \mathcal{A}_2|\mathcal{E}) = P(\epsilon|E_1)P(F_1, F_2, F_3|E_2)$$

This value should be fairly low, since the alignment does not map the English sentences to their translations. Hence, if our translation model is accurate we will have

$$P(\mathcal{F}, \mathcal{A}_1|\mathcal{E}) \gg P(\mathcal{F}, \mathcal{A}_2|\mathcal{E})$$

In general, the more sentences that are mapped to their translations in an alignment $\mathcal{A}$, the higher the value of $P(\mathcal{F}, \mathcal{A}|\mathcal{E})$. We can extend this idea to produce an alignment algorithm given a translation model. In particular, we take the alignment of a corpus $(\mathcal{E}, \mathcal{F})$ to be the alignment $\mathcal{A}$ that maximizes $P(\mathcal{F}, \mathcal{A}|\mathcal{E})$. The more accurate the translation model, the more accurate the resulting alignment will be.

However, because the parameters are all of the form $P(F_p|E)$ where E is a sentence, the above framework is not amenable to the situation where a French sentence corresponds to no English sentences. Hence, we use a slightly different framework. We view a bilingual corpus as a sequence of *sentence beads* (Brown *et al.*, 1991b), where a sentence bead corresponds to an irreducible group of sentences that align with each other. For example, the correct alignment of the bilingual corpus in Figure 2 consists of the sentence bead $[E_1; F_1]$ followed by the sentence bead $[E_2; F_2, F_3]$. We can represent an alignment $\mathcal{A}$ of a corpus as a sequence of sentence beads $([E_p^1; F_p^1], [E_p^2; F_p^2], \ldots)$, where the E_p^i and F_p^i can be zero, one, or more sentences long.

Under this paradigm, instead of expressing the translation model as a conditional distribution $P(F_p|E)$ we express the translation model as a distribution $P([E_p; F_p])$ over sentence beads. The alignment problem becomes discovering the alignment $\mathcal{A}$ that maximizes the joint distribution $P(\mathcal{E}, \mathcal{F}, \mathcal{A})$. Assuming that successive sentence beads are generated independently, we get

$$P(\mathcal{E}, \mathcal{F}, \mathcal{A}) = p(L) \prod_{k=1}^{L} P([E_p^k; F_p^k])$$

	English ($\mathcal{E}$)		French ($\mathcal{F}$)
E_1	`That is what the consumers are interested in and that is what the party is interested in.`	F_1	`Voilà ce qui intéresse le consommateur et voilà ce qui intéresse notre parti.`
E_2	`Hon. members opposite scoff at the freeze suggested by this party; to them it is laughable.`	F_2	`Les députés d'en face se moquent du gel que a proposé notre parti.`
		F_3	`Pour eux, c'est une mesure risible.`

Figure 2: A Bilingual Corpus

where $\mathcal{A} = ([E_p^1; F_p^1], \ldots, [E_p^L; F_p^L])$ is consistent with $\mathcal{E}$ and $\mathcal{F}$ and where $p(L)$ is the probability that a corpus contains L sentence beads.

2.2 The Basic Translation Model

For our translation model, we desire the simplest model that incorporates lexical information effectively. We describe our model in terms of a series of increasingly complex models. In this section, we only consider the generation of sentence beads containing a single English sentence $E = e_1 \cdots e_n$ and single French sentence $F = f_1 \cdots f_m$. As a starting point, consider a model that assumes that all individual words are independent. We take

$$P([E;F]) = p(n)p(m)\prod_{i=1}^{n} p(e_i) \prod_{j=1}^{m} p(f_j)$$

where $p(n)$ is the probability that an English sentence is n words long, $p(m)$ is the probability that a French sentence is m words long, $p(e_i)$ is the frequency of the word e_i in English, and $p(f_j)$ is the frequency of the word f_j in French.

To capture the dependence between individual English words and individual French words, we generate English and French words in pairs in addition to singly. For two words e and f that are mutual translations, instead of having the two terms $p(e)$ and $p(f)$ in the above equation we would like a single term $p(e, f)$ that is substantially larger than $p(e)p(f)$. To this end, we introduce the concept of a *word bead*. A word bead is either a single English word, a single French word, or a single English word and a single French word. We refer to these as 1:0, 0:1, and 1:1 word beads, respectively. Instead of generating a pair of sentences word by word, we generate sentences bead by bead, using the 1:1 word beads to capture the dependence between English and French words.

As a first cut, consider the following "model":

$$P^*(B) = p(l)\prod_{i=1}^{l} p(b_i)$$

where $B = \{b_1, \ldots, b_l\}$ is a multiset of word beads, $p(l)$ is the probability that an English sentence and a French sentence contain l word beads, and $p(b_i)$ denotes the frequency of the word bead b_i. This simple model captures lexical dependencies between English and French sentences.

However, this "model" does not satisfy the constraint that $\sum_B P^*(B) = 1$; because beadings B are unordered multisets, the sum is substantially less than one. To force this model to sum to one, we simply normalize by a constant so that we retain the qualitative aspects of the model. We take

$$P(B) = \frac{p(l)}{N_l}\prod_{i=1}^{l} p(b_i)$$

While a beading B describes an *unordered* multiset of English and French words, sentences are in actuality *ordered* sequences of words. We need to model word ordering, and ideally the probability of a sentence bead should depend on the ordering of its component words. For example, the sentence *John ate Fido* should have a higher probability of aligning with the sentence *Jean a mangé Fido* than with the sentence *Fido a mangé Jean*. However, modeling word order under translation is notoriously difficult (Brown *et al.*, 1993), and it is unclear how much improvement in accuracy a good model of word order would provide. Hence, we model word order using a uniform distribution; we take

$$P([E;F], B) = \frac{p(l)}{N_l n! m!}\prod_{i=1}^{l} p(b_i)$$

which gives us

$$P([E;F]) = \sum_B \frac{p(l_B)}{N_{l_B} n! m!} \prod_{i=1}^{l_B} p(b_i)$$

where B ranges over beadings consistent with $[E;F]$ and l_B denotes the number of beads in B. Recall that n is the length of the English sentence and m is the length of the French sentence.

2.3 The Complete Translation Model

In this section, we extend the translation model to other types of sentence beads. For simplicity, we only consider sentence beads consisting of one English sentence, one French sentence, one English sentence and one French sentence, two English sentences and one French sentence, and one English sentence and two French sentences. We refer to these as 1:0, 0:1, 1:1, 2:1, and 1:2 sentence beads, respectively.

For 1:1 sentence beads, we take

$$P([E;F]) = p_{1:1} \sum_B \frac{p_{1:1}(l_B)}{N_{l_B,1:1} n! m!} \prod_{i=1}^{l_B} p(b_i)$$

where B ranges over beadings consistent with $[E;F]$ and where $p_{1:1}$ is the probability of generating a 1:1 sentence bead.

To model 1:0 sentence beads, we use a similar equation except that we only use 1:0 word beads, and we do not need to sum over beadings since there is only one word beading consistent with a 1:0 sentence bead. We take

$$P([E]) = p_{1:0} \frac{p_{1:0}(l)}{N_{l,1:0} n!} \prod_{i=1}^{l} p(e_i)$$

Notice that $n = l$. We use an analogous equation for 0:1 sentence beads.

For 2:1 sentence beads, we take

$$Pr([E_1, E_2; F]) = p_{2:1} \sum_B \frac{p_{2:1}(l_B)}{N_{l_B,2:1} n_1! n_2! m!} \prod_{i=1}^{l_B} p(b_i)$$

where the sum ranges over beadings B consistent with the sentence bead. We use an analogous equation for 1:2 sentence beads.

3 Implementation

Due to space limitations, we cannot describe the implementation in full detail. We present its most significant characteristics in this section; for a more complete discussion please refer to (Chen, 1993).

3.1 Parameterization

We chose to model sentence length using a Poisson distribution, *i.e.*, we took

$$p_{1:0}(l) = \frac{\lambda_{1:0}^l}{l!\ e^{\lambda_{1:0}}}$$

for some $\lambda_{1:0}$, and analogously for the other types of sentence beads. At first, we tried to estimate each λ parameter independently, but we found that after training one or two λ would be unnaturally small or large in order to specifically model very short or very long sentences. To prevent this phenomenon, we tied the λ values for the different types of sentence beads together. We took

$$\lambda_{1:0} = \lambda_{0:1} = \frac{\lambda_{1:1}}{2} = \frac{\lambda_{2:1}}{3} = \frac{\lambda_{1:2}}{3} \qquad (1)$$

To model the parameters $p(L)$ representing the probability that the bilingual corpus is L sentence beads in length, we assumed a uniform distribution.[1] This allows us to ignore this term, since length will not influence the probability of an alignment. We felt this was reasonable because it is unclear what *a priori* information we have on the length of a corpus.

In modeling the frequency of word beads, notice that there are five distinct distributions we need to model: the distribution of 1:0 word beads in 1:0 sentence beads, the distribution of 0:1 word beads in 0:1 sentence beads, and the distribution of all word beads in 1:1, 2:1, and 1:2 sentence beads. To reduce the number of independent parameters we need to estimate, we tied these distributions together. We assumed that the distribution of word beads in 1:1, 2:1, and 1:2 sentence beads are identical. We took the distribution of word beads in 1:0 and 0:1 sentence beads to be identical as well except restricted to the relevant subset of word beads and normalized appropriately, *i.e.*, we took

$$p_e(e) = \frac{p_b(e)}{\sum_{e' \in B_e} p_b(e')} \quad \text{for } e \in B_e$$

and

$$p_f(f) = \frac{p_b(f)}{\sum_{f' \in B_f} p_b(f')} \quad \text{for } f \in B_f$$

[1] To be precise, we assumed a uniform distribution over some arbitrarily large finite range, as one cannot have a uniform distribution over a countably infinite set.

where p_e refers to the distribution of word beads in 1:0 sentence beads, p_f refers to the distribution of word beads in 0:1 sentence beads, p_b refers to the distribution of word beads in 1:1, 2:1, and 1:2 sentence beads, and B_e and B_f refer to the sets of 1:0 and 0:1 word beads in the vocabulary, respectively.

3.2 Evaluating the Probability of a Sentence Bead

The probability of generating a 0:1 or 1:0 sentence bead can be calculated efficiently using the equation given in Section 2.3. To evaluate the probabilities of the other sentence beads requires a sum over an exponential number of word beadings. We make the gross approximation that this sum is roughly equal to the maximum term in the sum. For example, with 1:1 sentence beads we have

$$\begin{aligned} P([E;F]) &= p_{1:1} \sum_B \frac{p_{1:1}(l_B)}{N_{l_B,1:1} n! m!} \prod_{i=1}^{l_B} p(b_i) \\ &\approx p_{1:1} \max_B \{ \frac{p_{1:1}(l_B)}{N_{l_B,1:1} n! m!} \prod_{i=1}^{l_B} p(b_i) \} \end{aligned}$$

Even with this approximation, the calculation of $P([E;F])$ is still intractable since it requires a search for the most probable beading. We use a greedy heuristic to perform this search; we are not guaranteed to find the most probable beading. We begin with every word in its own bead. We then find the 0:1 bead and 1:0 bead that, when replaced with a 1:1 word bead, results in the greatest increase in probability. We repeat this process until we can no longer find a 0:1 and 1:0 bead pair that when replaced would increase the probability of the beading.

3.3 Parameter Estimation

We estimate parameters by using a variation of the Viterbi version of the *expectation-maximization* (EM) algorithm (Dempster *et al.*, 1977). The Viterbi version is used to reduce computational complexity. We use an incremental variation of the algorithm to reduce the number of passes through the corpus required.

In the EM algorithm, an *expectation* phase, where counts on the corpus are taken using the current estimates of the parameters, is alternated with a *maximization* phase, where parameters are re-estimated based on the counts just taken. Improved parameters lead to improved counts which lead to even more accurate parameters. In the incremental version of the EM algorithm we use, instead of re-estimating parameters after each complete pass through the corpus, we re-estimate parameters after each sentence. By re-estimating parameters continually as we take counts on the corpus, we can align later sections of the corpus more reliably based on alignments of earlier sections. We can align a corpus with only a single pass, simultaneously producing alignments and updating the model as we proceed.

More specifically, we initialize parameters by taking counts on a small body of previously aligned data. To estimate word bead frequencies, we maintain a count $c(b)$ for each word bead that records the number of times the word bead b occurs in the most probable word beading of a sentence bead. We take

$$p_b(b) = \frac{c(b)}{\sum_{b'} c(b')}$$

We initialize the counts $c(b)$ to 1 for 0:1 and 1:0 word beads, so that these beads can occur in beadings with nonzero probability. To enable 1:1 word beads to occur in beadings with nonzero probability, we initialize their counts to a small value whenever we see the corresponding 0:1 and 1:0 word beads occur in the most probable word beading of a sentence bead.

To estimate the sentence length parameters λ, we divide the number of word beads in the most probable beading of the initial training data by the total number of sentences. This gives us an estimate for $\lambda_{1:0}$, and the other λ parameters can be calculated using equation (1).

We have found that one hundred sentence pairs are sufficient to train the model to a state where it can align adequately. At this point, we can process unaligned text and use the alignments we produce to further train the model. We update parameters based on the newly aligned text in the same way that we update parameters based on the initial training data.[2]

[2] In theory, one cannot decide whether a particular sentence bead belongs to the best alignment of a corpus until the whole corpus has been processed. In practice, some partial alignments will have much higher probabilities than all other alignments, and it is desirable to train on these partial alignments to aid in aligning later sections of the corpus. To decide when it is reasonably safe to train on a particular sentence bead, we take advantage of the thresholding described in Section 3.4, where improbable partial

To align a corpus in a single pass the model must be fairly accurate before starting or else the beginning of the corpus will be poorly aligned. Hence, after bootstrapping the model on one hundred sentence pairs, we train the algorithm on a chunk of the unaligned target bilingual corpus, typically 20,000 sentence pairs, before making one pass through the entire corpus to produce the actual alignment.

3.4 Search

It is natural to use dynamic programming to search for the best alignment; one can find the most probable of an exponential number of alignments using quadratic time and memory. Alignment can be viewed as a "shortest distance" problem, where the "distance" associated with a sentence bead is the negative logarithm of its probability. The probability of an alignment is inversely related to the sum of the distances associated with its component sentence beads.

Given the size of existing bilingual corpora and the computation necessary to evaluate the probability of a sentence bead, a quadratic algorithm is still too profligate. However, most alignments are one-to-one, so we can reap great benefits through intelligent thresholding. By considering only a subset of all possible alignments, we reduce the computation to a linear one.

Dynamic programming consists of incrementally finding the best alignment of longer and longer prefixes of the bilingual corpus. We prune all alignment prefixes that have a substantially lower probability than the most probable alignment prefix of the same length.

3.5 Deletion Identification

Deletions are automatically handled within the standard dynamic programming framework. However, because of thresholding, we must handle large deletions using a separate mechanism.

Because lexical information is used, correct alignments receive vastly greater probabilities than incorrect alignments. Consequently, thresholding is generally very aggressive and our search beam in the dynamic programming array is narrow. However, when there is a large deletion in one of the parallel corpora, consistent lexical correspondences disappear so no one alignment has a much higher probability than the others and our search beam becomes wide. When the search beam reaches a certain width, we take this to indicate the beginning of a deletion.

To identify the end of a deletion, we search linearly through both corpora simultaneously. All occurrences of words whose frequency is below a certain value are recorded in a hash table. Whenever we notice the occurrence of a rare word in one corpus and its translation in the other, we take this as a candidate location for the end of the deletion. For each candidate location, we examine the forty sentences following the occurrence of the rare word in each of the two parallel corpora. We use dynamic programming to find the probability of the best alignment of these two blocks of sentences. If this probability is sufficiently high we take the candidate location to be the end of the deletion. Because it is extremely unlikely that there are two very similar sets of forty sentences in a corpus, this deletion identification algorithm is robust. In addition, because we key off of rare words in considering ending points, deletion identification requires time linear in the length of the deletion.

4 Results

Using this algorithm, we have aligned three large English/French corpora. We have aligned a corpus of 3,000,000 sentences (of both English and French) of the Canadian Hansards, a corpus of 1,000,000 sentences of newer Hansard proceedings, and a corpus of 2,000,000 sentences of proceedings from the European Economic Community. In each case, we first bootstrapped the translation model by training on 100 previously aligned sentence pairs. We then trained the model further on 20,000 sentences of the target corpus. Note that these 20,000 sentences were not previously aligned.

Because of the very low error rates involved, instead of direct sampling we decided to estimate the error of the old Hansard corpus through comparison with the alignment found by Brown of the same corpus. We manually inspected over 500 locations where the two alignments differed to estimate our error rate on the alignments disagreed upon. Taking the error rate of the Brown alignment to be 0.6%, we estimated the overall error

alignments are discarded. At a given point in time in aligning a corpus, all undiscarded partial alignments will have some sentence beads in common. When a sentence bead is common to all active partial alignments, we consider it to be safe to train on.

rate of our alignment to be 0.4%.

In addition, in the Brown alignment approximately 10% of the corpus was discarded because of indications that it would be difficult to align. Their error rate of 0.6% holds on the remaining sentences. Our error rate of 0.4% holds on the entire corpus. Gale reports an approximate error rate of 2% on a different body of Hansard data with no discarding, and an error rate of 0.4% if 20% of the sentences can be discarded.

Hence, with our algorithm we can achieve at least as high accuracy as the Brown and Gale algorithms *without* discarding any data. This is especially significant since, presumably, the sentences discarded by the Brown and Gale algorithms are those sentences most difficult to align.

In addition, the errors made by our algorithm are generally of a fairly trivial nature. We randomly sampled 300 alignments from the newer Hansard corpus. The two errors we found are displayed in Figures 3 and 4. In the first error, E_1 was aligned with F_1 and E_2 was aligned with F_2. The correct alignment maps E_1 and E_2 to F_1 and F_2 to nothing. In the second error, E_1 was aligned with F_1 and F_2 was aligned to nothing. Both of these errors could have been avoided with improved sentence boundary detection. Because length-based alignment algorithms ignore lexical information, their errors can be of a more spectacular nature.

The rate of alignment ranged from 2,000 to 5,000 sentences of both English and French per hour on an IBM RS/6000 530H workstation. The alignment algorithm lends itself well to parallelization; we can use the deletion identification mechanism to automatically identify locations where we can subdivide a bilingual corpus. While it required on the order of 500 machine-hours to align the newer Hansard corpus, it took only 1.5 days of real time to complete the job on fifteen machines.

5 Discussion

We have described an accurate, robust, and fast algorithm for sentence alignment. The algorithm can handle large deletions in text, it is language independent, and it is parallelizable. It requires a minimum of human intervention; for each language pair 100 sentences need to be aligned by hand to bootstrap the translation model.

The use of lexical information requires a great computational cost. Even with numerous approximations, this algorithm is tens of times slower than the Brown and Gale algorithms. This is acceptable given that alignment is a one-time cost and given available computing power. It is unclear, though, how much further it is worthwhile to proceed.

The natural next step in sentence alignment is to account for word ordering in the translation model, *e.g.*, the models described in (Brown *et al.*, 1993) could be used. However, substantially greater computing power is required before these approaches can become practical, and there is not much room for further improvements in accuracy.

Acknowledgements

We give special thanks to Peter Brown, Stephen DellaPietra, Vincent DellaPietra, and Robert Mercer for their suggestions, support, and relentless taunting. We thank Jan Hajic and Meredith Goldsmith as well as the aforementioned for checking the alignments produced by the implementation. In addition, Stuart Shieber and Andrew Kehler provided valuable comments on earlier drafts of this paper.

The preparation of this paper was supported in part by Presidential Young Investigator grant IRI-9157996 to Stuart M. Shieber and matching grants from the Digital Equipment Corporation and the Xerox Corporation.

References

(Bellman, 1957) Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton N.J., 1957.

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

(Brown *et al.*, 1991a) Peter F. Brown, Stephen A. DellaPietra, Vincent J. DellaPietra, and Robert L. Mercer. Word sense disambiguation using statistical methods. In *Proceedings 29th Annual Meeting of the ACL*, pages 265–270, Berkeley, CA, June 1991.

(Brown *et al.*, 1991b) Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. Aligning sentences in parallel corpora. In *Proceedings 29th Annual Meeting of the ACL*, pages 169–176, Berkeley, CA, June 1991.

E_1	`If there is some evidence that it ... and I will see that it does.`	F_1	`Si on peut prouver que elle ... je verrais à ce que elle se y conforme. \SCM{} Language = French \ECM{}`
E_2	`\SCM{} Translation \ECM{}`	F_2	`\SCM{} Paragraph \ECM{}`

Figure 3: An Alignment Error

E_1	`Motion No. 22 that Bill C-84 be amended in ... and substituting the following therefor : second anniversary of.`	F_1	`Motion No 22 que on modifie le projet de loi C-84 ... et en la remplaçant par ce qui suit : ‘ 18.`
		F_2	`Deux ans après : ’.`

Figure 4: Another Alignment Error

(Brown *et al.*, 1993) Peter F. Brown, Stephen A. DellaPietra, Vincent J. DellaPietra, and Robert L. Mercer. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 1993. To appear.

(Catizone *et al.*, 1989) Roberta Catizone, Graham Russell, and Susan Warwick. Deriving translation data from bilingual texts. In *Proceedings of the First International Acquisition Workshop*, Detroit, Michigan, August 1989.

(Chen, 1993) Stanley F. Chen. Aligning sentences in bilingual corpora using lexical information. Technical Report TR-12-93, Harvard University, 1993.

(Dagan *et al.*, 1991) Ido Dagan, Alon Itai, and Ulrike Schwall. Two languages are more informative than one. In *Proceedings of the 29th Annual Meeting of the ACL*, pages 130–137, 1991.

(Dempster *et al.*, 1977) A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(B):1–38, 1977.

(Gale and Church, 1991) William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting of the ACL*, Berkeley, California, June 1991.

(Gale *et al.*, 1992) William A. Gale, Kenneth W. Church, and David Yarowsky. Using bilingual materials to develop word sense disambiguation methods. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 101–112, Montréal, Canada, June 1992.

(Kay, 1991) Martin Kay. Text-translation alignment. In *ACH/ALLC '91: "Making Connections" Conference Handbook*, Tempe, Arizona, March 1991.

(Klavans and Tzoukermann, 1990) Judith Klavans and Evelyne Tzoukermann. The bicord system. In *COLING-90*, pages 174–179, Helsinki, Finland, August 1990.

(Sadler, 1989) V. Sadler. *The Bilingual Knowledge Bank – A New Conceptual Basis for MT.* BSO/Research, Utrecht, 1989.

(Warwick and Russell, 1990) Susan Warwick and Graham Russell. Bilingual concordancing and bilingual lexicography. In *EURALEX 4th International Congress*, Málaga, Spain, 1990.