

# BILINGUAL SENTENCE ALIGNMENT: BALANCING ROBUSTNESS AND ACCURACY

Michel Simard  
*simard@citi.doc.ca*

Pierre Plamondon  
*plamondo@citi.doc.ca*

Centre for Information Technology Innovation (CITI)  
1575 Chomedey Blvd.  
Laval (Quebec)  
CANADA H7V 2X2

## Abstract

*Sentence alignment* is the problem of making explicit the relations that exist between the sentences of two texts that are known to be mutual translations. Automatic sentence alignment methods typically face two kinds of difficulties. First, there is the question of *robustness*. In real life, discrepancies between the source-text and its translation are quite common: differences in layout, omissions, inversions, etc. Sentence alignment programs must be ready to deal with such phenomena. Then, there is the question of *accuracy*. Even when translations are "clean", alignment is still not a trivial matter: some decisions are hard to make, even for humans. We report here on the current state of our ongoing efforts to produce a sentence alignment program that is both robust and accurate. The method that we propose relies on two new alignment engines, and combines the robustness of so-called "character-based" methods with the accuracy of stochastic translation models. Experimental results are presented, that demonstrate the method's effectiveness, and highlight where problems remain to be solved.

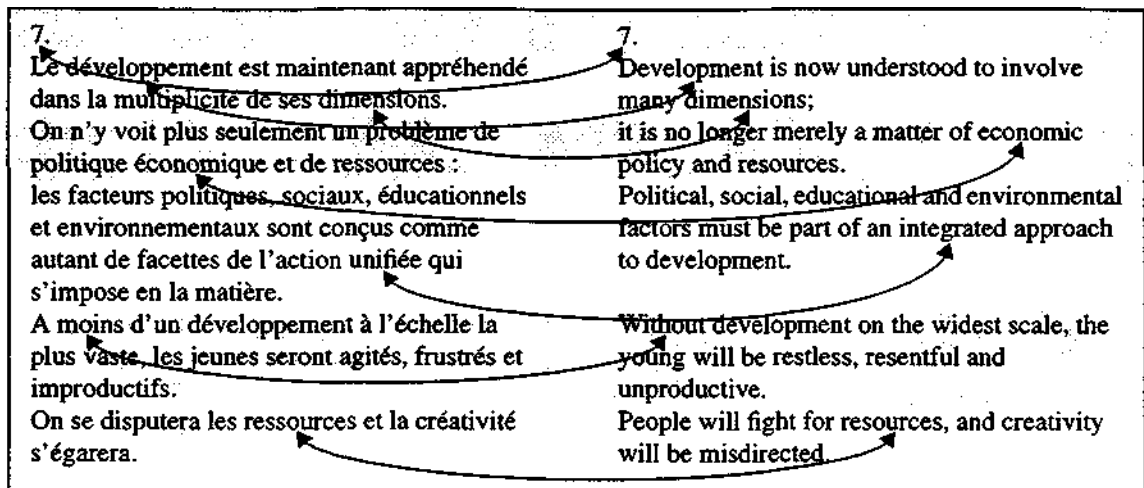
## Introduction

The bitext correspondence problem (*BCP*) can be loosely described as that of making explicit the relations that exist between two texts that are known to be mutual translations. The result of this operation can take many forms, but the output of most existing bitext correspondence methods falls into one of two categories:

- An *alignment* is a parallel segmentation of the two texts, typically into small logical units such as sentences, such that the  $n$ th segment of the first text and the  $n$ th segment of the second are mutual translations.
- A *bitext map* is a set of pairs  $(x, y)$ , where  $x$  and  $y$  refer to precise locations in the first and second texts respectively, with the intention of denoting portions of the texts that correspond to one another.

This is illustrated in figure 1.

Bitext correspondences are of vital interest to anyone who wishes to exploit existing translations as an active source of information. Which is best between an alignment and a bitext map usually depends on the intended application. By definition, an alignment covers the totality of the bitext. In this sense, it is both *exhaustive* and *exact*: for each segment of text, it says something like "the translation of this segment is exactly that segment". The same cannot be said of bitext maps. Some methods, such as those proposed by Church [5] or Fung and McKeown [7], produce approximate maps (i.e. not exact), that say something like "The translation of the text around this point is somewhere around that point". Other methods, such as those



**Figure 1: Alignment vs. bitext map – alternating grey and white areas denote the segments of an alignment; arrows denote one possible bitext map.**

proposed by Dagan *et al.* [6] or Melamed [12] produce maps that are exact ("the translation of the object at position  $x$  is the object at position  $y$ ") but not exhaustive. On the other hand, what they lack in exactness or exhaustiveness, bitext map usually make up for in resolution: they give a "closer view" on the correspondence.

There are many situations where alignments are preferable, however. In particular, this appears to be true of applications where the bitext correspondence is directly intended for a human. An example of such an application is the bilingual concordance system developed at CITI [16]. This system allows a user to query a large corpus of bitext for specific expressions in one or both languages. Most often, the purpose of the user is to find out how a given expression is translated. Using a sentence alignment for such a system has two advantages: first, given that exhaustive and accurate mappings at the level of expressions are not yet available, it ensures that the excerpts of bitext returned by the system contain both the queried expression and its translation; second, it allows the system to return the expression and its translation within a coherent context, so that the user can evaluate the relevance of each returned item with regard to his own problem.

In our ongoing efforts to develop sentence alignment methods that are both reliable and accurate, we have developed a hybrid approach, that combines the best of existing methods. This work is described in the following pages.

## Background

As far as we know, interest in bitext correspondence began sometime in the mid-eighties, at which time independent efforts were being pursued concurrently in many places, most notably at Xerox PARC [11], IBM's Thomas J. Watson research centre [2], AT&T Bell Laboratories in Murray Hill [8] and in Geneva, at ISSCO [3]. Interestingly, all these early efforts focussed on sentence alignments rather than bitext maps.

The first communications on the subject were published in 1991. Paradoxically, there were two of them, they appeared back-to-back in the proceedings of the Conference of the Association for Computational Linguistics, and described alignment methods that were virtually identical. Both were based on a statistical modelization of translations that only took into account the length of the text segments (sentences, paragraphs), and relied on a dynamic programming scheme to find the most likely alignment. The main difference between the two approaches was how length was measured: while Brown *et al.* [2] counted words, Gale and Church [8] counted characters.

The early successes obtained using these methods almost gave the impression that the problem had been solved. Of course, this was not the case, and although it is true that sentence alignment is mostly an easy problem, anyone who has attempted to manually align a sufficient amount of text knows that there are situations where even humans have a hard time making a decision. The truth of the matter is that BCP is just one instance of the more general *translation analysis* problem (see [9]), which turns out to be "AI-complete". In other words, to solve the BCP entirely, you would first have to solve all the other "hard" AI problems - and conversely, if you solve the BCP, then you have just put the whole AI community out of work!

What is it that is so difficult with sentence alignment?

The first issue is that of *robustness*. For a long time, almost everybody in the field was working with the same set of data, namely the Hansards (Canadian parliamentary proceedings). As other multilingual corpora became available, it quickly appeared that the Hansards were exceptionally "clean" translations. As Church points out, "Real texts are noisy" ([5]). Earlier methods are likely to wander off track when faced with deviations from the standard "linear" progression of translation, such as those that occur when parts of the source text do not make their way into the translation (omissions), or end up in a different order (inversions).

To deal with the robustness issue, Church took a very straightforward and intuitive approach, exploiting an alignment criterion that was first proposed by Simard et al [15]: cognate words. Cognates are pairs of words of different languages that have close etymological ties. Often this tie will be reflected both in the meanings and orthography of these words. As a result, they are likely mutual translations, and they are fairly easy to detect, even for someone who is not familiar with either of the languages involved. Church's program, called `char_align`, does not rely on a formal definition of cognates, but rather on a more general notion of "resemblance" between source-text and translation. Interestingly, what `char_align` does could very well be compared to what a human would do to get a rough bitext mapping, i.e. take a certain distance from the texts, and look for similarities in layout, or for obvious clues such as numbers, proper names and so on.

Unlike its predecessors, `char_align` will usually not be fooled by omissions, inversions and other oddities. At the same time, again for the sake of robustness, the program does not rely on an *a priori* segmentation of the texts into paragraphs and sentences. Church realized that this was one of the major problems with earlier approaches: proper segmentation is not a trivial problem, and incorrect segmentations are bound to lead to incorrect alignments. He resolved the problem by building a program that completely ignores logical text divisions. (As a matter of fact, `char_align` is not even interested in words - what it aligns are *bytes*). Also, because an alignment that is not based on the texts' logical divisions does not seem to make much sense, the program produces a bitext map.

The second issue is that of *accuracy*: even when the input texts are "clean", alignment programs are sometimes faced with hard decisions. In order to obtain the best possible alignments, one will eventually have to throw in the whole armada of NLP and AI techniques: dictionaries, grammars, semantic networks, stochastic language models, common-sense reasoning, intelligent agents - you name it.

So far, the most promising avenues in dealing with this problem make use of stochastic translation models. For example, to compute sentence alignments, Chen [4] replaces the simple length-based models of earlier methods by a more elaborate model that takes into account the words of the text. Dagan *et al* [6] use a similar model to obtain word-level mappings.

To this day, most research on the BCP has focussed on either one of these two problems (robustness and accuracy). This work is an attempt to tackle the two together.

## **Robust and Accurate Sentence Alignments**

We now describe our approach to the sentence alignment problem. Our idea is to combine the robustness of "character-based" methods, such as `char_align`, and the accuracy of stochastic translation models. This

idea is implemented as a two-step strategy: first compute a bitext map, working on robustness rather than accuracy; then use this map to constraint the search-space for the computation of the sentence alignment, this time relying on a method that favors accuracy over robustness or efficiency.

### ***First Step: Initial Bitext Mapping***

The initial bitext map is computed using a program that we call *Jacal* ("just another cognate alignment program"), which was itself inspired by Melamed's SIMR program [12]. What *Jacal* does is match *isolated cognates*:

- We consider two word-forms of different language to be cognates if their four first characters are identical, disregarding letter-case or diacritics. In spite of its simplicity, this operational definition of cognates works well for related pairs of languages such as French and English, as demonstrated by Simard *et al.* [15]
- We consider an occurrence of a word-form to be *isolated* if no occurrence of *resembling* word-forms appear within a certain window around this occurrence. This *isolation window* is measured in characters, and is set to cover a given fraction of the text considered, say 30%.
- As for the notion of *resemblance* between word-forms, it is exactly identical to that of *cognateness*, except that it applies to pairs of word-forms of the same language.

To explain how *Jacal* determines which pairs of *isolated cognates* should be matched, it is convenient to look at bitext maps from a graphical point of view, as if both texts to map were respectively laid out along the X and Y axes in the plane (see figure 2 below).

*Jacal* initially includes two points in the map: those that correspond to the beginnings and ends of the texts. Assuming that the alignment is going to lie somewhere along the line segment that connects these two points, it draws this line, and then a "corridor" around it, whose width is proportional to the distance between the two initial points. It then adds to the set only those points corresponding to pairs of isolated cognates that lie within the corridor.

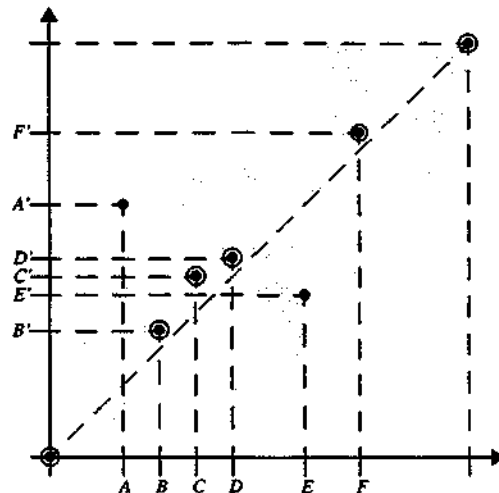
Now, while most of the points found using this method are true correspondences, some may be wrong. We have found that most of these erroneous points are easy to detect: they are usually "not in line" with their neighbors. To eliminate these points, *Jacal* relies on a simple smoothing technique, based on linear regression.

Of course, our matching criterion is quite strict, and very few word-pairs are actually selected, so that the bitext map is very sparse. But because it is also extremely reliable, we can now repeat the process: successively take as *anchors* each consecutive pair of points already in the map, disregard all surrounding text, and apply the same method between anchors, i.e. find isolated cognates along the search corridor, and then smooth out rogue points. *Jacal* applies this process recursively until no more points can be found.

Once this is done, we have found it useful to apply a final, two-pass smoothing. The first pass is identical to what is done during the recursive search: it gets rid of aberrations that sometimes appear when the final result is pieced together. The second pass is based on the simple observation that "isolated" points in the map, say those that are more than 150 characters away from their closest neighbors, are often wrong, even if they are in line with those neighbors. So we just eliminate them.

### ***Intermediate Step: Segmentation and Search-space Determination***

The next question is: How can we use the output of *Jacal* to obtain an accurate sentence-based alignment? There are two very distinct aspects to this problem: the first has to do with segmenting the text into



**Figure 2: Jacal cognate matching method – the grayed region is the search corridor, circled points are those that belong to the map; point (A, A') is out because it falls outside the search corridor; point (E, E') is out because it is not in line with its neighbors (smoothing)**

sentences, the second with determining the search-space, i.e. the pairs of sentences that will be considered for alignment.

As pointed out earlier, an incorrect segmentation of the text into sentences is bound to lead to incorrect alignments. In fact, analogous comments can be made about many other natural language analysis applications. Paradoxically, only recently has the problem been addressed seriously by the NLP research community.

For the time being, we rely on a rather simplistic method for segmentation, based almost exclusively on language-independent data, essentially a set of rules encoding general knowledge about the structure of electronic texts. Notable exceptions are language-specific lists of abbreviations and acronyms, which are used to determine whether a period following a word belongs to the word itself or serves to end a sentence (we ignore the possibility of a period simultaneously serving both purposes).

As for the task of determining the search-space for the final alignment, it consists in deciding which sentences can be paired, and which ones cannot. One way of doing this is to look at a sentence alignment as a special case of a bitext map, i.e. one where the mapped points are constrained to coincide with sentence-boundaries. We can assume that the points of the correct sentence-alignment will lie not too far from the points produced by Jacal. We have also observed that when the Jacal points are dense, then we are likely to find the correct sentence-alignment points close by. Conversely, as the Jacal points get scarcer, we have to widen our search area.

In practice, what we do is take each pair of adjacent points of the initial mapping and draw a hexagonal-shaped "corridor" around the points, i.e. a rectangle with its corners cut off. The width of each corridor is proportional to the distance between the two points it connects. We then include in the search-space all pairs of sentence-boundaries that fall within these regions. This is illustrated in figure 3.

The resulting set of points constitutes the search-space for the final sentence alignment: it determines exactly those points where the bitext may be segmented. Interestingly, the bitext maps generated by Jacal are generally quite dense (typically, one point for every 5-6 words of each text), so that for most sentence-boundaries of one text, there is only one possible match in the other.

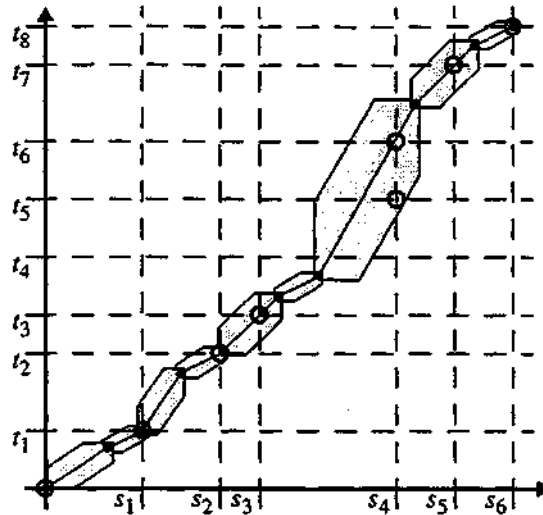


Figure 3: Computing the search-space from a bitext map (black dots) - overlapping search regions are drawn between each consecutive pair of points (grayed areas), whose width is proportional to the distance between the points; only those pairs of sentence-boundaries that fall within these regions are included (circled intersections).

### *Second step: Final Sentence Alignment*

From this point on, any sentence alignment program that is capable of working within such a restricted search-space can be used to finish up the job. Following the ideas of Chen [4] and Dagan *et al.* [6], we have developed a method that could probably be referred to as "heavy artillery" in this context: it is based on a *statistical lexical translation model*, namely Brown *et al.*'s "Model 1"[1]. Essentially, the model consists in a set of parameters  $T_{fe}$ , that estimate the probability of observing word  $f$  in one text, given that word  $e$  appears in the other. The parameters are normally estimated from frequencies observed in a large collection of pairs of text segments known to be mutual translations (typically, these segments are sentences).

The parameters of the model can be combined so as to estimate the probability of observing some arbitrary set of words in one language, given some other set in the other language. In particular, this may be applied to estimate how likely it is to observe one sentence given another one.

Clearly, such a model can also be used to score competing sentence alignment hypothesis. This is precisely what our program, called *Salign*, does. Using a dynamic programming scheme similar to those used in previous sentence alignment programs, *Salign* finds the alignment with the maximum overall probability.

Most alignment methods that make use of lexical information assume that this information is not available *a priori*: Kay and Röscheisen [11], Fung and McKeown [7], Chen [4], Dagan *et al.* [6] all go to great lengths to infer the parameters of their models directly from the pair of texts to align. This is a very interesting approach, especially when dealing with many language pairs. But for language pairs such as English and French, for which large quantities of aligned bitext already exist, it seems a little bit like re-inventing the wheel every time. Furthermore, with such methods, aligning short texts can become a problem. *Salign* normally assumes the existence of a trained model - in fact, its implementation allows it to deal with large models, covering vocabularies of tens of thousands of word-forms. For example, the model we used was trained on approximately three years of Hansard proceedings. However, nothing in the method precludes a bootstrapping approach, which could easily be implemented using the initial bitext map.

# Evaluation

We have produced an implementation of the sentence alignment method described in the previous section. In order to assess its performance, we needed two things: first, a corpus of text for which a "reference" sentence alignment exists, i.e. an alignment reputed to be "correct"; second, some way of measuring how the output of our program differed from the reference. We feel that this last aspect has been somewhat neglected in previous work, which makes it very hard to compare methods, or simply to know what to expect from a given program.

The corpus we used is the *BAF* corpus (see [13]): this is a collection of French-English bitexts, hand-aligned to the sentence level. The corpus consists in a dozen pairs of text files, totalling a little over 400 000 words in each language. Most of the texts are of an "institutional" nature (Hansards, UN reports, etc.), but the corpus also contains scientific, technical and literary material.

All documents of the corpus were split in two more or less equal parts. The first halves (the "training" corpus) were used for the purpose of optimizing the various parameters of the program, while the second halves (the "test" corpus) were kept for computing the final results<sup>1</sup>.

Performance was measured using a method based on a metric proposed by Pierre Isabelle [10]: Consider two texts,  $S$  and  $T$ , viewed as unordered sets of sentences:  $S = \{s_1, s_2, \dots, s_n\}$  and  $T = \{t_1, t_2, \dots, t_m\}$ . An alignment  $A$  may be represented as a subset of  $S \times T$ :

$$A = \{ (s_1, t_1), (s_2, t_2), (s_2, t_3), \dots, (s_n, t_m) \}$$

with the interpretation that  $(s_i, t_j) \in A$  if and only if  $s_i$  and  $t_j$  share a common clause (in the above example, the fact that  $s_2$  appears in two couples simply means that  $s_2$  is translated partly by  $t_2$  and partly by  $t_3$ ). What we need is a way of measuring the difference between some alignment  $A$  and a "correct" reference alignment  $A_R$ . Borrowing from the information retrieval terminology, we define:

- **Recall:**  $|A_R \cap A| / |A_R|$
- **Precision:**  $|A_R \cap A| / |A|$

From the end-user's point of view, these notions of recall and precision can be loosely interpreted like this: Say you are examining some region  $s$  of text in  $S$  and the corresponding region  $t$  in  $T$ , as given by some alignment  $A$ . *Recall* denotes your chances of finding the translation of  $s$  in  $t$ , while *precision* refers to the proportion of  $t$  that is actually related to  $s$ .

After using this measure for some time, we realized that it was somewhat unfair to the alignment programs, because most alignment errors occurred on small sentences. To correct this situation, we used a variant of this method, where recall and precision are measured in terms of *characters* rather than sentences. When formulated this way, there is also a graphical interpretation to these notions, which is illustrated in figure 4.

In order to better understand the effect of each component of our program, we designed a number of experiments, the results of which are summarized in table 1 below. First, as a point of comparison, the test-corpus was submitted to the Simard *et al.* (*SFI*) program. The same texts were then submitted to a "Jacal / Gale and Church" (*J+GC*) combination. The aim of this first experiment was to evaluate how using a Jacal bitext map could improve the robustness of a length-based approach. The comparison with SFI is also interesting, because in a sense, they represent "opposite" strategies: while SFI tries to improve length-based alignments using cognates, the J+GC combo proceeds the other way around.

---

1. One of the BAF bitexts, a technical manual, had to be discarded from the corpus, because it contained in appendix a relatively large glossary of terms, sorted alphabetically. Since the order of the entries was completely different in French and English, no attempt was made to hand-align this glossary. Therefore, the reference alignment contained one very large pair of segments. The presence of this region distorted the performance measures to the point where they were no longer significant.

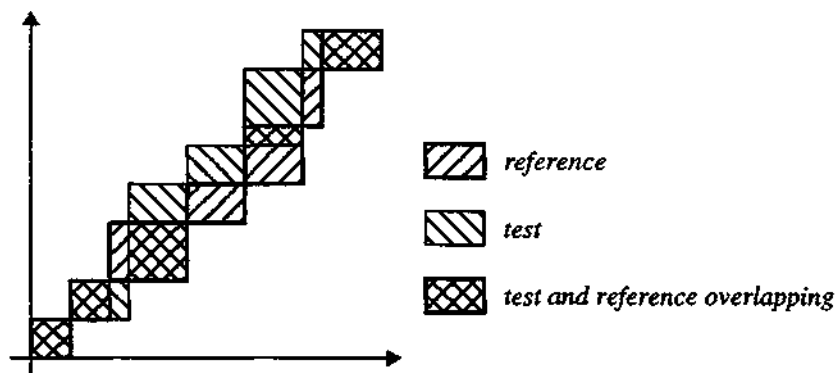


Figure 4: Graphical interpretation of alignment *recall* and *precision* — An alignment may be seen as a set of square regions in the plane; *recall* denotes how much of the reference regions are covered by the test regions; *precision* denotes how much of the test regions overlaps with the reference regions.

As can be seen in table 1, using the output of Jacal to guide a length-based alignment technique clearly improves alignment recall. In some cases the improvement is minor, but there are situations where this makes the difference between an alignment that is literally beyond repair, and one that is acceptable (for example, the first scientific article). The situation with alignment precision is not as clear, however: although J+GC is more precise than SFI on average, the opposite appears to be true when the "harder" texts are discarded.

We then ran the Salign program on the test-corpus, with the intent of seeing how this program would do "on its own" (in this setup, Salign operated on a fixed-width window along the "diagonal" that joins the beginnings and ends of the two texts). Here again, the result is a general improvement on alignment recall, both over the SFI and J+GC programs. As for alignment precision, it is approximately the same as for the J+GC combo.

Finally, we ran Jacal and Salign together on the test-corpus. Interestingly, the results are exactly the same as those obtained with Salign alone, except for the pair of literary texts, where using Jacal to guide the search significantly improved both recall and precision. This pair of texts (Jule Vernes' *De la terre à la lune*) is particularly interesting, because it shows how a translation can sometimes diverge from the original. (In fact, in this case, it is not even clear whether the English version is indeed a translation of the French, or if it was based on an abridged version.)

## Conclusions

We have described our attempt to develop a method for aligning sentences that is both robust and accurate. Both the Jacal and Salign programs have been implemented in C, and are quite efficient (although it must be said that Salign typically requires a fairly large amount of memory to run). As an example, we recently used the Jacal / Gale and Church combination to align eight years of Hansard proceedings (approximately 70 million words in all) for our translation memory application. The process only took about four hours on a Sun Spare Ultra I. As far as we know, using Salign instead of the Gale and Church program would not have been much more costly in time.

As far as robustness is concerned, the results are very encouraging: our method was able to satisfyingly align all texts of the BAF corpus, even the "harder" ones. As for accuracy, however, it would appear to remain a problem. In fact, one thing that may come as a surprise is how the overall results are poor, regardless of the program used. Performance levels below 95% are not exactly what the literature on the subject had us used to. It could be the case that this is just a consequence of our choice of performance metric. On the other



Document	Text Size (words)		Simard et al.		Jacal + Gale & Church		Salign		Jacal + Salign	
	<i>English</i>	<i>French</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>
Hansard	27654	29511	97.39	99.23	97.42	97.54	99.06	98.30	99.06	98.30
UN Sec.-Gen. Annual Report	24536	27850	96.79	99.08	97.03	98.47	97.23	98.35	97.23	98.35
UN Int'l Labour Org. Report	74414	75523	90.14	96.54	93.06	97.64	92.70	98.46	92.70	98.46
Court Transcripts	15741	16508	92.40	97.30	90.26	94.88	92.94	95.81	92.94	95.81
	6344	6299	10.40	15.59	88.07	81.55	89.47	83.47	89.39	83.97
	9527	10397	83.62	98.55	84.45	96.81	85.25	96.82	85.25	96.82
Scientific Articles	3293	3858	95.65	89.60	95.50	87.08	96.42	84.44	96.42	84.44
	3556	3551	91.94	97.01	94.44	97.01	93.19	96.97	93.19	96.97
	1667	1687	95.15	98.79	96.29	92.28	98.29	94.04	98.29	94.04
Literary	19597	26725	39.97	39.06	72.19	57.66	87.70	46.09	93.99	54.49
Total / Average	186329	201909	83.52	87.98	90.99	92.09	93.20	91.28	93.92	92.26

**Table 1: Alignment results for various programs on individual texts of the BAF corpus. Recall and precision values are given as percentages; average values are weighted by text size.**

hand, the figures obtained seem to confirm one of our earlier claims: that the Hansards are exceptionally easy to align when compared to other text genres.

The low precision levels obtained with our methods highlight one of the current shortcomings of Salign: the model on which it is based is unable to account for omissions or additions in a translation. As a result, source segments that do not find their way into the translation are absorbed by neighboring segments in the alignment, thus reducing precision. We are currently investigating various solutions to this problem.

Another thing that we realized when examining our alignment errors is that many of them are actually the result of segmentation errors. In some cases, this can have a dramatic effect on recall and precision measures: typically, "over-segmentation" reduces alignment recall, while "under-segmentation" reduces alignment precision. It should be noted, however, that for an application such as bilingual concordance, such errors are not necessarily catastrophic. From our experience, alignment errors resulting from over-segmentation usually separate unrelated portions of sentences, while under-segmentation simply results in a "dilution" of the information rather than in genuine misalignments. We are nevertheless exploring the possibility of using more sophisticated segmentation methods, such as those proposed by Palmer and Hearst [14] for disambiguating periods. However, it would seem that ambiguous periods is not the only issue at stake here. In fact, most of our problems come from "sentences" that simply do not end with a period, or any punctuation mark for that matter: titles, section headings, list and table items, etc. What remains to be demonstrated, however, is that plain ASCII text such as those that we have been working with, are really what real-life texts will be like in the near future...

## Acknowledgments

We would like to thank Pierre Isabelle and Dan Melamed, for inspirational discussions and constructive comments.

## References

- [1] Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer (1993), "The Mathematics of Machine Translation: Parameter Estimation", in *Computational Linguistics*, Vol. 19, No 2.
- [2] Brown, Peter, J. Lai and R. Mercer (1991), "Aligning Sentences in Parallel Corpora", in *Proceedings of ACL-91*.
- [3] Catizone, Roberta, Graham Russell and Susan Warwick (1989), "Deriving Translation Data from Bilingual Texts", in *Proceedings of the First International Lexical Acquisition Workshop*, Detroit.
- [4] Chen, Stanley F. (1993), "Aligning Sentences in Bilingual Corpora Using Lexical Information", in *Proceedings of ACL-93*, Columbus OH.
- [5] Church, Kenneth W. (1993), "Char\_align: A Program for Aligning Parallel Texts at the Character Level", in *Proceedings of ACL-93*, Columbus OH.
- [6] Dagan, Ido, Kenneth W. Church and William A. Gale (1993), "Robust Bilingual Word Alignment for Machine Aided Translation", in *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*.
- [7] Fung, Pascale and Kathleen McKeown (1994), "Aligning Noisy Parallel Corpora Across Language Groups: Word Pair Feature by Dynamic Time Warping", in *Proceedings of AMTA-94*.
- [8] Gale, William A. and Kenneth W. Church (1991), "A Program for Aligning Sentences in Bilingual Corpora", in *Proceedings of ACL-91*.
- [9] Isabelle, Pierre *et al.* (1993), "Translation Analysis and Translation Automation", in *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation*, Kyoto, Japan.
- [10] Isabelle, Pierre (1996), *personal communication*.
- [11] Kay, Martin and Martin Röscheisen (1993), "Text-Translation Alignment", in *Computational Linguistics*, Vol. 19, No. 1.
- [12] Melamed, I. Dan (1996), "A Geometric Approach to Mapping Bitext Correspondence", to appear in the *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia.
- [13] Macklovitch, Elliott *et al.* (1996), *BAF: un corpus de bi-texte anglais-français annoté à la main*, to appear.
- [14] Palmer, David D. and Marti A. Hearst (1994), *Adaptive Sentence Boundary Disambiguation*, Report No. UCB/CSD 94/797, Computer Science Division (EECS), University of California, Berkeley.
- [15] Simard, Michel, George F. Foster and Pierre Isabelle (1992), "Using Cognates to Align Sentences in Bilingual Corpora", in *Proceedings of TMI-92*.
- [16] Simard, Michel, George F. Foster, Francois Perrault (1993), *TransSearch: un concordancier bilingue*, CITI Technical Report.