Cognates Alignment

Paper-ID: ACL-2001-0081

Abstract

Some authors (Michel Simard *et al.*; Dan Melamed; Pernilla Danielsson and Katarina Mühlenbock) have suggested measures of similarity of words in different languages so as to find extra clues for alignment of parallel texts. Cognate words, like 'Parliament' and 'Parlement', in English and French respectively, provide extra anchors that help to improve the quality of the alignment. In this paper, we will extend an alignment algorithm proposed by António Ribeiro *et al.* using typical contiguous and non-contiguous sequences of characters extracted using a statistically sound method (Gaël Dias *et al.*). With these typical sequences, we are able to find more reliable correspondence points and improve the alignment quality without recurring to heuristics to identify cognates.

1 Introduction

Alignment of parallel texts (texts which are mutual translations) is one of the first steps to be taken to build automatically a database of translation equivalents for bilingual lexicography or cross-lingual text processing tasks, such as machine(–aided) translation, cross-language information retrieval, multilingual question–answering systems to name but a few applications. Thus, it becomes crucial that those parallel texts should be as closely aligned as possible. That is to say, we should be able to make as detailed correspondences as possible between passages of texts and their translations in the other languages. Much work has already been done on sentence alignment, from early work by Peter Brown *et al.* (1991), William Gale and Kenneth Church (1991) and Martin Kay and Martin Röscheisen (1993), to alignment of smaller text segments as in Michel Simard *et al.* (1992), Kenneth Church (1993), Pascale Fung and Kathleen McKeown (1997), Dan Melamed (1999) and António Ribeiro *et al.* (2000a,b).

Some methods have relied on using similar words, namely cognates (e.g. 'Parliament' and 'Parlement', in English and French respectively), in order to get extra clues for alignment. Several measures of "cognateness" have been suggested (Michel Simard *et al.*, 1992; Dan Melamed 1999; Pernilla Danielsson and Katarina Mühlenbock, 2000) but none is sufficiently reliable. That is, they do not provide any statistical studies supporting them and are tailored for specific applications.

In this paper, we will extend a method of alignment proposed by António Ribeiro *et al.* (2000a,b) by using typical contiguous and non-contiguous sequences of characters identified by statistical data analysis as shown in Gaël Dias *et al.* (2000b).

We will start by giving an overview of several heuristics that have been proposed so far in order to identify cognates. In section 3, we will describe the methodology used to identify typical contiguous and non-contiguous sequences of characters and, in section 4, the alignment algorithm is presented. An evaluation of the results is given in section 5 and, finally, we will draw some conclusions and present some future work.

2 Previous Work

In order to make the most of word similarities in parallel texts for alignment, some attempts have been made to use *cognates*. According to the Longman Dictionary of Applied–Linguistics, a *cognate* is "a word in one language which is similar in form and meaning to a word in another language because both languages are related" (Jack Richards *et al.*, 1985, p. 43). For example, 'Parliament' and 'Parlement', in English and French respectively, are cognates.

When two words have the same or similar forms in two languages but have different meanings in each of them, they are called *false cognates* or *false friends* (Jack Richards *et al.*, 1985, p. 43). For example, the English word 'library' and the French word 'librairie' are false

cognates (Dan Melamed, 1999, p. 114). 'library' translates as 'bibliothèque' in French and, conversely, 'librairie' as 'bookstore' in English.

Michel Simard *et al.* (1992) was the first to propose exploiting cognates for alignment. They considered two words as cognates if their first four characters were identical (Michel Simard *et al.*, 1992, p 71), as in 'Parliament' and 'Parlement'. This simple heuristic proved to be quite useful, providing a greater number of points of correspondence though it has some shortcomings. According to this rule, the English word 'government' and the French word 'gouvernement' are not cognates. Also, 'conservative' and 'conseil' (*council*) are cognates: different word endings are not distinguished.

In order to exploit this similarity in words, Dan Melamed (1999, p. 113) proposed a "more accurate cognate criterion" driven by approximate string matching. Dan Melamed suggested a similarity measure between two tokens based on the longest common sub-sequence of shared characters. For example, for the case of the 'government' and 'gouvernement', the longest common sub-sequence happens to be 'government', the same as the English word. The sub-sequence does not have to be necessarily contiguous but it must keep the same character order. Dan Melamed proposed the Longest Common Sub-sequence Ratio as:

$$\frac{length(Longest\ Common\ Sub\text{-}Sequence(w_1, w_2))}{max(length(w_1), length(w_2))}$$

Equation 1. The longest common sub-sequence ratio between words w_1 and w_2.

This measure gives the ratio of the length of the longest common sub-sequence and the length of the longest token. For the last example, the ratio is 10 (the length of 'government') over 12 (the length of 'gouvernement') whereas for 'conservative' and 'conseil', the ratio is just 6 over 12. It tends to favour long sequences similar to the longest word and to penalise sequences which are too short compared to a long word.

For the alignment purposes, Dan Melamed selects all pairs of words which have a ratio above a certain threshold. However, and again, this is just another heuristic which seems to provide better results than the one first proposed by Michel Simard *et al.* (1992) but without a statistical supporting study.

Pernilla Danielsson and Katarina Mühlenbock (2000) aim at aligning cognates starting from aligned sentences in two quite similar languages: Norwegian and Swedish. The "fuzzy match" of two words is "calculated as the number of matching consonants[,] allowing for one mismatched character" (Pernilla Danielsson and Katarina Mühlenbock, 2000, p. 162). For example, the Norwegian word 'plutselig' (*suddenly*) and the Swedish word 'plötsligt' would be matched by 'pltslg': all consonants match except for one ('t'). However, 'bakspeilet' (*rear-view mirror*) and 'backspegeln', in Norwegian and Swedish respectively, would not match because four consonants are not shared ('c', 'g', 'n', 't').

In this paper, we propose not to use any of these heuristics to identify cognates. Instead, we shall say that if two sequences of characters are *typical* for a pair of languages, then their level of "cognateness" is quite high. In other words, two words are candidate cognates if they *share* a *typical* sequence of characters that is common to that pair of languages. These typical sequences of characters are extracted using a statistical measure as described in the next section. For example, the English word 'Government' and the Portuguese word 'governo' share a sequence of characters that is typical of both languages: '•_overn' (the dot '•' stands for the character space and the underscore for any character). Another example is the character sequence '•pe_so_s•' as in 'pessoas' and 'persons'.

3 Extraction of Cognates

Before starting the alignment, we must identify typical sequences of characters common to both languages. In this section we will give an overview of the method used for extracting them.

3.1 Source Parallel Corpora

For this experiment we tested the extraction of typical sequences of characters and alignment on three pairs of languages: Portuguese–English (henceforth, pt-en), Portuguese–French (pt-fr) and Portuguese–Spanish (pt-es).

The parallel corpora consists of judgements of the The Court of Justice of the European Communities[1]. We chose five judgements at random translated in the four languages. For each language, it amounts to 15k words (about 80k characters) with an average of 5 pages per text. This corresponds to about 3k words per text (15k characters per text).

[1] Webpage address: `http://curia.eu.int`

3.2 The Method of Extraction

From the linguistic point of view, cognates are words that show in the similarity of their forms that they derive from a common parent. Thus, both words 'government' in English and 'gouvernement' in French would be considered cognates. Michel Simard *et al.* (1992) go even further in the definition of cognates considering them as "pairs of tokens of different languages which share "obvious" phonological or orthographic and semantic properties, with the result that they are likely to be used as mutual translations". Thus, cognates are recognised on the fly according to a series of rules. For example, Kenneth Church (1993) used the rule of identical 4-grams to find an alignment path between the source and the target language texts.

However, very few dedicated researches have been dealing with the specific objective of identifying and extracting cognates in parallel texts. As mentioned above, many application-specific methodologies have been proposed but none has ever been evaluated outside the considered application.

In order to overcome the lack of a unified methodology, we propose an original way to identify cognates based on the notion of character association. We strongly believe that cognates are recurrent and highly cohesive sequences of characters that are common to two or more languages. As a consequence, cognates may be considered as specific character associations that can be identified by statistical data analysis as shown in (Gaël Dias *et al.*, 2000b). In this context, we use a statistically-based architecture called SENTA (Software for the Extraction of N-ary Textual Associations) that retrieves contiguous and non-contiguous textual associations from real texts. As defined in (Gaël Dias *et al.*, 2000a), SENTA can be divided into three main steps, each one evidencing relevant improvements in the domain of extractors:

1. Segmentation of the input text into positional n-grams of text units, for n≥2;
2. Evaluation of the degree of cohesiveness of each n-gram using the Mutual Expectation association measure; and,
3. Extraction of candidate text associations by using the GenLocalMaxs algorithm.

In this algorithm the cohesion measure of a n-gram must be greater than the cohesion of all the n–1 grams contained in it and greater than the cohesion of all the n+1 grams which contain the n-gram.

Candidate cognates[2] should be extracted by SENTA from the mixture of text corpora in different languages in order to get the typical character sequences common to those languages. This situation is illustrated in Figure 1, where L1 and L2 stand for any two different languages.

We used the parallel corpora presented in the previous sub-section. For each pair of languages, we fed SENTA with the respective set of parallel texts in order to extract the typical sequences of characters for that specific pair.

As a result of the extraction process, SENTA builds a list of potentially relevant multilingual character associations together with their Mutual Expectation score (measure of cohesiveness) and frequency.

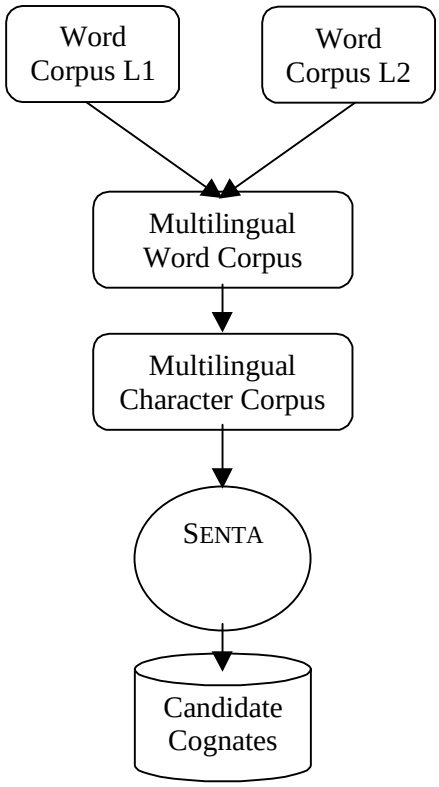

Figure 1: Extraction Process

At this point, three important remarks need to be stressed out. First, SENTA allows the extraction of typical non-contiguous sequences of characters, thus allowing the identification of cognates that do not embody continuous strings, as in the method proposed by Dan Melamed (1999). Consequently, a cognate like 'At_mic' is identified, subsuming both the English word

[2] Statistical methodologies cannot guarantee that the extracted character associations are true cognates. As a consequence, we will denote them as candidates.

'Atomic' and the Portuguese word 'Atómico'. Second, cognates of any length can be identified unlike most approaches that propose four characters as a magic number. Third, candidate cognates are supported by numerical values that give some important clues about their pertinence.

4 Alignment

After identifying the typical contiguous and non-contiguous character sequences, we proceed to the alignment of the parallel texts. It is only at this stage that it is possible to confirm whether two candidate typical character sequences found in the parallel texts are true cognates.

4.1 Background

We will use an alignment algorithm based on the work reported in António Ribeiro *et al.* (2000 a,b). This algorithm is based on the fact that words tend to occur in similar positions in parallel texts. They tend to appear along a diagonal of a rectangle whose sides are proportional to the sizes of each text (see the figure below). Those points that do not fit, end up being removed using statistically supported filters.

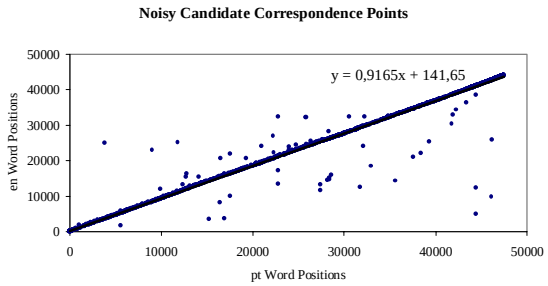

Figure 2: Alignment of parallel texts using word positions. Around the diagonal several noisy correspondence points can be seen. The equation of the linear regression line formed by all correspondence points is shown on the top.

Basically, the algorithm starts by pairing the positions of words which are identical in two languages and which occur with equal frequencies in parallel pieces of text. For example, suppose the word 'Euratom' occurs three times in one parallel Portuguese–English text. Suppose it is the 228th, 620th and 3016th word in the Portuguese text and it is the 202th, 577th and 2771th word in the English text. Then, three correspondence points would be defined using those word positions: (228,202), (620,577) and (3016,2771).

However, not all correspondence points defined in this way are "well-behaved" as Figure 2 shows. Sometimes, this method makes wrong pairings of words which lead to the noisy points around the diagonal as shown in the figure. That is, the method may pair words which are too distant from their expected positions (somewhere near the diagonal determined by the linear regression of the correspondence points).

False friends could be a cause of concern for this approach. For example, the Portuguese word 'embaraçada' (*embarrassed*) and the Spanish word 'embarazada' (*pregant*) are false cognates. Since they have such different meanings they appear in different contexts, in different parts of the text. Thus, associating them would produce a noisy correspondence point which would end up being filtered out.

The algorithm proposes the use of a statistical filter based on confidence bands of linear regression lines in order to reject noisy points of correspondence. Since the algorithm is recursive, it is able to explore reliable correspondence points within each aligned parallel piece of text.

In our case, we are looking not only for identical words but also for typical contiguous and non-contiguous character sequences in the texts of two languages. Moreover, these sequences do not necessarily start where a word starts like the case of the sequence '•_overn', which match with 'Government' and 'governo', or the sequence 'itua__o' which matches with the end of the words 'situation' and 'situação'. Consequently, we can no longer take words as the smallest text unit. We must work at character level instead.

For this reason, the alignment algorithm must be adapted for character alignment. In particular, it had to be adapted to handle the matching of typical character sequences at each character position in the parallel texts. For these experiments, we extracted character sequences from four to seven characters long.

Table 1: Number of typical sequences of characters for each pair of languages.

Pair	Typical Sequences
pt-en	677
pt-es	1137
pt-fr	877

Bearing in mind that Portuguese and Spanish are two quite similar languages, it does not come as a surprise to see that this pair of languages has more typical sequences of character than any of the other pairs. French comes next since for its closeness as a romance language and English

comes last confirming the fact that Portuguese and English are more distant languages.

4.2 Indexing

The most computationally expensive task lies actually before the alignment proper. That was one of the reasons that led us to start with small texts. The amount of data processed for these experiments corresponds to more than 300k characters.

First, all texts need to be indexed. For an average sized text of 15k characters (3k words), the current implementation of the indexer takes about 30 minutes on a Pentium II 366MHz with 64MB.

```
818 the          |  577 •s_b_e
821 •_urope      |  578 sobre
822 European     |  584 a
824 rope__•      |  585 •__ter
830 •At_mi       |  585 •_n_er
831 Atomic       |  585 •inte
837 •Energ       |  586 interpretação
838 Energy       |  587 nte_pr
844 •Com         |  600 do
845 Community    |  602 •_rti
848 muni         |  603 artigo
855 and          |  610 4
```

Figure 3: Indexing words and typical sequences of characters in two parallel texts in English and Portuguese. Several sequences may start in the same position. The numbers show the character or *byte* position in the file.

The character position of each word and of each typical character sequence needs to be recorded. The figure above shows an example. The indexer needs to check if the token is a word on its own or if it matches any of the extracted candidate cognates.

4.3 Character Alignment

Secondly, we proceed to the alignment proper. Since we no longer have correspondence points built from word numbers, we had to introduce a new concept based on the position of a typical character sequence. Instead of using the position of the median character of a token (Dan Melamed, 1999, p. 108) or the median position of a typical character sequence, we decided to use the position of the first and last characters of a sequence of characters as the correspondence points. These two points create a segment which we shall call a *segment of correspondence*. This segment delimits the anchor used in each parallel text. Figure 4 gives an example.

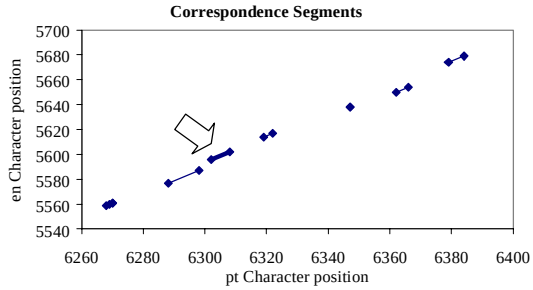

Figure 4: Each of the segments shown in this figure correspond to the beginning and end of a word or a typical character sequence which has been paired. The arrow points to the segment defined by the sequence '•_overn'.

The segments in this figure were built from the co-ordinates of the paired sequences of characters (the anchors). For example, the sequence '•_overn' which helps to make the correspondence between the words 'Government' and 'Governo' defines the segment shown in Figure 4, with co-ordinates (6302,5596) (6308,5595).

```
6268|10                  |  5559|10
6270|.                   |  5561|.
6271| ¶        Na        |  5562|     In the
nova                     |  New
6288| declaração          |  5577| Declaratio
(•_eclara__o)            |  (•_eclara__o)
6299| do                  |  5588|n by the
6302| Govern              |  5596| Govern
(•_overn)                |  (•_overn)
6309|o do Reino           |  5603|ment of the
6319| Uni                 |  5614| Uni
6323|do da Grã-           |  5618|ted Kingdom of
Bretanha e da            |  Great
```

Figure 5: Alignment of a Portuguese–English parallel text. Segments of correspondence are in bold. The numbers correspond to character positions. The typical character sequences are shown inside brackets.

We should note that some segments may result from merging overlapping segments. That is a common result when one word has several typical character sequences. For example, in Figure 5, the sequence '•_eclara__o' results from merging the sequences '•_eclar', 'clara' and 'lara__o' which were found to be typical of both English and Portuguese by the extractor of candidate cognates, though the underlying word is different. In this case, the cognate was clearly identified. Furthermore, these sequences may happen to span across several words, linking some of them. For example, the sequence '•li_re•circula__o' for the pair Portuguese–French subsumes both the Portuguese expression 'livre circulação' (*free movement*) and the French translation 'libre circulation'. This longer character sequence results from merging several short typical character

several short typical character sequences: '•li_re', 'i_re ci', '•circ', 'i_cula', 'rc_la' and 'cula__o'. In the end, even though we did not start with long typical character sequences, we are able to use the small ones and merge them as they overlap.

For the alignment algorithm, we need to distinguish between two sets of segments of correspondence: the candidates and the final segments. The former set provides a possible set of correspondences (or anchors) between the parallel texts. The latter, refers to the set of correspondences which leads to the alignment.

Here is an overview of the algorithm.

1. Take two parallel texts A and B;
2. For each text, build a table with the character positions of each word and each typical sequence of characters;
3. Define the texts' beginnings – the point (0,0) – and the texts' ends – the point (length of text A, length of text B) – as the extremes of the initial search rectangle;
4. Build a set of candidate segments of correspondence
   4.1. Consider as candidates those defined by identical sequences of characters (either words or typical characters sequences) which occur with the same frequency within the search rectangle;
   4.2. Define the extremes of the segment from the co-ordinates of the beginning and of the end of the common character sequence;
5. Filtering out bad points
   5.1. Build a linear regression line using the co-ordinates of each candidate segment;
   5.2. Filter out the extreme points using the histogram of distances between expected and real positions of each point (António Ribeiro *et al*., 2000 a,b);
   5.3. Filter out points which lie outside the confidence bands of the linear regression line (António Ribeiro *et al*., 2000 a,b);
6. For each of the candidate segment of correspondence, check if both extreme points were selected as good points of the linear regression; otherwise, remove the segment from the set of candidate segments of correspondence since it has unreliable points;
7. For each of the selected candidate segments of correspondence, merge those which overlap;
8. Add all the remaining candidate segments to the set of final segments of correspondence;
9. For each new segment of correspondence, repeat steps 4 to 9 (recursive algorithm) to the search space defined by the end of the last segment of correspondence and the beginning of the next segment of correspondence.

After repeating these steps, we get a set of segments of correspondence which link the anchors in both parallel texts. Moreover, we get true cognates in the segments of correspondence.

5 Evaluation

The most computationally expensive tasks for this approach lie on the extraction of typical character sequences and on the indexing of the texts according to the positions of words and of typical character sequences. The alignment proper, on a Pentium II 366MHz with 64MB, takes about 5 minutes for a 30k characters text (the largest in the set of parallel texts).

We compared our results with the results obtained from a recursive algorithm reported in António Ribeiro *et al.* (2000a) that does not use cognates. The table below shows the results:

Table 2: Comparison of the average number of segments and the average number of characters in each aligned text segment without using cognates (António Ribeiro *et al.*, 2000a) and using cognates.

	Without cognates		With cognates	
Pair	#Segments	#Characters per Aligned Segment	#Segments	#Characters per Aligned Segment
pt-en	754	18,7	988	13
pt-es	1264	13,4	1446	8
pt-fr	1012	15,9	1353	9
Average	1010	16,0	1263	10

If we compare the ratios of the number of segments obtained and the ratios of the sizes of each aligned segment, we can see that using cognates leads to a significant improvement in the alignment. By sizes of aligned segments, we mean the number of characters found between two consecutive segments of correspondence (between two anchors).

Table 3: Comparison of the ratios of the number of segments and the size of each aligned segment.

	Ratios	
Pair	#Segments	#Characters per Aligned Segment
pt-en	+31%	-29%
pt-es	+14%	-39%
pt-fr	+34%	-44%
Average	+25%	-37%

Table 3 shows that the size of each segment was reduced by almost 40% with an increase of 25% of the number of segments. The figure below shows the histogram of the sizes of the segments for the pair Portuguese–English.

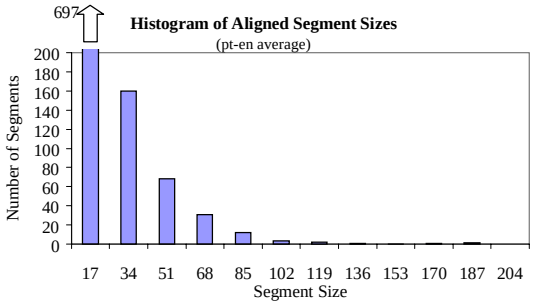

Figure 6: Average Size of the Aligned segment sizes. Most of the segments have less than 50 characters for the Portugues–English parllel texts.

6 Conclusions

In this paper we have presented a method to align parallel texts that uses both identical words and typical contiguous and non-contiguous character sequences extracted using a statistically sound method (Gaël Dias *et al.*, 2000a,b). This method provides a first level statistical support that was not yet available for identifying candidate cognates. The alignment itself confirms the "cognateness" of two text segtypical character sequences.

Typical character sequences help to identify cognates in parallel texts that can be used as anchors for alignment purposes. They form segments of correspondence delimited by the positions of the beginning and of the end of each sequence of characters. They are filtered using a methodology described in António Ribeiro *et al.* (2000 a,b) and adapted for this case of alignment at character level.

However, considering characters as the smallest text unit instead of using words increased the complexity of the alignment algorithm. Nonetheless, the results have proven that it is possible to improve the alignment results, reducing by almost 40% the size of each small piece of aligned text. In this way, we are able to have a finer grained alignment. Moreover, this strategy is not limited to pairing words: it is able to work above word level as long as typical character sequences span across several words.

7 Future Work

We intend to apply this methodology to larger texts in order to confirm our results. All in all, we believe it will bring much better alignments. This will allow us to extract translation equivalents more reliably using a methodology similar to the one described by António Ribeiro *et al.* (2000c). Also, we want to extract multiword units translations. We will start by considering them as textual units and, combining with the approach presented in this paper, it will allow us to make better pairings of similar multiword units.

The approach reported in this paper also opens new avenues of research for Asian languages: it provides a means of handling alignment of parallel text of languages in which it is difficult to find word boundaries as it is the case of some Asian languages, like Chinese and Japanese. It becomes possible to bootstrap the alignment algorithm using typical sequences of characters which are common to a pair of languages.

References

Peter Brown, Jennifer Lai and Robert Mercer. 1991. *Aligning Sentences in Parallel Corpora*. In "Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics", Berkeley, California, U.S.A., pp. 169–176

Kenneth Church. 1993. *Char_align: A Program for Aligning Parallel Texts at the Character Level*. In "Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics", Columbus, Ohio, U.S.A., pp. 1–8

Pernilla Danielsson and Katarina Mühlenbock. 2000. *Small but Efficient: The Misconception of High-Frequency Words in Scandinavian Translation*. In John White (ed.), "Envisioning Machine Translation in the Information Future – Proceedings of the 4th Conference of the Association for Machine Translation in the Americas, AMTA 2000 – Lecture Notes in Artificial Intelligence", volume 1934, Springer-Verlag 2000, Berlin Heidelberg, pp. 158–168.

Gaël Dias, Sylvie Guilloré and Gabriel Lopes. 2000a. *Extraction Automatique d'Associations Textuelles à Partir de Corpora Non Traités.* In M. Rajman and J. Chapelier (eds.) "Actes des 5e Journées Internationales d'Analyse Statistique des Données Textuelles",

École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, pp. 213–221.

Gaël Dias, Sylvie Guilloré and Gabriel Lopes. 2000b. *Mining Textual Associations in Text Corpora*. In "Proceedings of the Workshop on Text Mining at the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", Boston, Massachusetts, USA.

ELRA (European Language Resources Association). 1997. *Multilingual Corpora for Cooperation*, Disk 2 of 2. Paris, France.

Pascale Fung and Kathleen McKeown. 1994. *Aligning Noisy Parallel Corpora across Language Groups: Word Pair Feature Matching by Dynamic Time Warping*. In "Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas", Columbia, Maryland, U.S.A., pp. 81–88.

Pascale Fung and Kathleen McKeown. 1997. *A Technical Word- and Term-Translation Aid Using Noisy Parallel Corpora across Language Groups*. Machine Translation, 12/1–2 (Special issue), pp. 53–87.

William Gale and Kenneth Church. 1991. *A Program for Aligning Sentences in Bilingual Corpora*. In "Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics", Berkeley, California, U.S.A., pp. 177–184 (short version). Also (1993) Computational Linguistics, 19/1, pp. 75–102 (long version).

Martin Kay and Martin Röscheisen. 1993. *Text-Translation Alignment*. Computational Linguistics, 19/1, pp. 121–142.

I. Dan Melamed. 1999. *Bitext Maps and Alignment via Pattern Recognition*. Computational Linguistics, 25/1, pp. 107–130.

António Ribeiro, Gabriel Lopes and João Mexia. 2000a. *Linear Regression Based Alignment of Parallel Texts Using Homograph Words*. In Werner Horn (ed.) (2000) "ECAI 2000: Proceedings of the 14th European Conference on Artificial Intelligence", volume 54, IOS Press 2000, Amsterdam, The Netherlands, pp. 446–450.

António Ribeiro, Gabriel Lopes and João Mexia. 2000b. *Using Confidence Bands for Parallel Texts Alignment*. In "Proceedings of the 38th Conference of the Association for Computational Linguistics (ACL 2000)", Association for Computational Linguistics 2000, pp. 432–439.

António Ribeiro, Gabriel Lopes and João Mexia. 2000c. *A Self-Learning Method of Parallel Texts Alignment*. In John White (ed.) (2000), "Envisioning Machine Translation in the Information Future – Proceedings of the 4th Conference of the Association for Machine Translation in the Americas, AMTA 2000 – Lecture Notes in Artificial Intelligence", volume 1934, Springer-Verlag 2000, Berlin Heidelberg, pp. 30–39.

Jack Richards, John Platt, Heidi Weber. 1985. *Longman Dictionary of Applied Linguistics*, Longman, UK.

Michel Simard, George Foster and Pierre Isabelle. 1992. *Using Cognates to Align Sentences in Bilingual Corpora*. In "Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation TMI-92", Montreal, Canada, pp. 67–81.

Michel Simard and Pierre Plamondon. 1998. *Bilingual Sentence Alignment: Balancing Robustness and Accuracy*. Machine Translation, 13/1, pp. 59–80.