Alignment of Paragraphs in Bilingual Texts using Bilingual Dictionaries and Dynamic Programming*

Alexander Gelbukh and Grigori Sidorov

Natural Language and Text Processing Laboratory,
Center for Research in Computer Science, National Polytechnic Institute,
Av. Juan Dios Batiz, s/n, Zacatenco, 07738, Mexico City, Mexico
www.Gelbukh.com; sidorov@cic.ipn.mx

Abstract. Parallel text alignment is a special type of pattern recognition task aimed to discover the similarity between two sequences of symbols. Given the same text in two different languages, the task is to decide which elements---paragraphs in case of paragraph alignment---in one text are translations of which elements of the other text. One of the applications is training training statistical machine translation algorithms. The task is not trivial unless detailed text understanding can be afforded. In our previous work we have presented a simple technique that relied on bilingual dictionaries but does not perform any syntactic analysis of the texts. In this paper we give a formal definition of the task and present an exact optimization algorithm for finding the best alignment.

1 Introduction

Given the same text in two different languages, the parallel text alignment task consists in deciding which elements of one text are translations of which one of the other text [9]. The task is useful in learning bilingual dictionaries and in training statistical machine translation algorithms. Viewed more generally as a pattern recognition task, the problem consists in identifying correspondences in two sequences of objects, which could be, say, text and speech or video recordings from different cameras [13][14]. While both the task and our suggested method are quite general, in this paper we concentrate on alignment of paragraphs in bilingual texts.

Various researchers have tried different approaches to text alignment, usually at sentence level [3][5][17], and a number of alignment tools are available.[1] Some methods rely on similarity between certain words in the two text—for example, words that are graphically similar can be considered pivots for rough alignment [18]. In a previous paper [8] we have suggested an alignment method based on measuring similarity

* Work done under partial support of Mexican Government (CONACyT, SNI) and National Polytechnic Institute, Mexico (CGPI, COFAA). We thank an anonymous reviewer for attracting our attention to valuable resources and publications.

[1] *Bilingual Sentence Aligner* by Robert C. Moore, research.microsoft.com/research/downloads/default.aspx; *Geometric Mapping and Alignment* (GMA) by Dan Melamed, nlp.cs.nyu.edu/GMA/; *Champollion Toolkit* by LDC, champollion.sourceforge.net/; an on-line sentence aligner, 143.107.183.175/site2001/projetos/pesa.htm.

using bilingual dictionaries and presented an approximate heuristic greedy alignment algorithm. In this paper our goals are:

- To formalize the paragraph alignment task, casting it as an optimization problem;
- To introduce an algorithm that finds the exact optimum of this problem, instead of the approximate heuristic-based algorithm;
- To suggest a distance measure for paragraphs that guarantees unbiased solution;
- To propose a baseline distance measure and to compare the results obtained with our suggested measure against such a baseline.

The optimization problem resulting from our formalization of the task strongly resembles string alignment problems, such as optimal string alignment or calculating the Levenshtein distance between strings. Inspired in standard methods for solving problems of this class, we developed a dynamic programming algorithm. However, our formalization differs from the optimal string alignment. That latter task requires aligning some symbols in a string with at most one symbol in the other string; in our case, we align every symbol with at least one symbol in the other string. This leads to a modification of the algorithm.

The paper is organized as follows. In Section 2, we explain the task in detail and formalize it as an optimization problem. In Section 3, we introduce a baseline and a suggested distance measures between paragraphs, which are used for calculation of the cost function to be optimized. In Section 4, we present a dynamic programming algorithm that finds the exact optimum of the problem. In Section 5, we discuss its complexity. Finally, in Section 6 we present the experimental results and in Section 7 give conclusions and discuss the possible future work.

2 Paragraph-Level Text Alignment

Given a text and its translation into another language, the text alignment task consists in determining which text elements (such as words) are translations of each other, as shown in Fig. 1, where words that are translations of each other are connected by lines. In such simple cases the text alignment task can be formalized as building a bipartite graph whose vertices are the text elements and an arc connects two vertices if the corresponding elements are translations of each other.

However, in more complex cases a whole set of text elements are translated by another set of elements, while this correspondence cannot be broken down into pair-

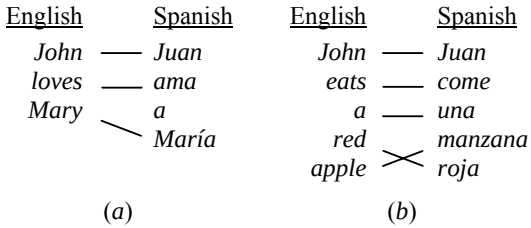

Fig. 1. Word-level alignment for the sentences *John loves Mary* and *John eats a red apple*.

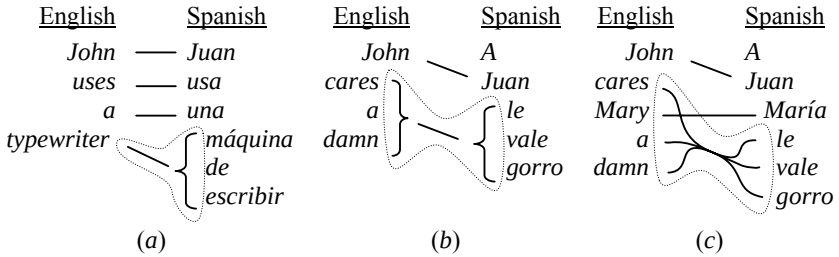

Fig. 2. Set-to-set alignment (literally: (*a*) 'device of writing', (*b*, *c*) 'is worth a cap to him'). The dotted lines mark single hyperarcs.

wise correspondences of individual elements, as shown in Fig. 2. In Fig. 2 (*a*), one word is translated by a whole run of words. In Fig. 2 (*b*), a whole run is translated by another run, while there is no pair-wise translation correspondences between the individual words in these two runs. Finally, Fig. 2 (*c*) shows the most general case: the correspondence holds between (non-contiguous) sets of elements.

While the structure shown in Fig. 1 can be formalized by a graph, the structure shown in Fig. 2 (*c*) is formalized by a generalization of the notion of a graph called *hypergraph*. Given a set of vertices **V**, a hypergraph **G** on **V** is defined as a graph whose vertices are non-empty subsets of **V**, a *hyperarc a* being a pair of subsets of **V**: $a = \{\mathbf{X}, \mathbf{Y}\}$, $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$, $\mathbf{X}, \mathbf{Y} \neq \varnothing$. A hyperarc can be graphically represented by a link with several "ends", as in Fig. 2 (*c*), or in a simplified form as a connection between grouped vertices, as in Fig. 2 (*a*, *b*). The bilingual text alignment task deals with *bipartite* hypergraphs. A (hyper)graph is called bipartite if its vertices are of two kinds: $\mathbf{V} = \mathbf{A} \cup \mathbf{B}$, $\mathbf{A} \cap \mathbf{B} = \varnothing$, and arcs connect elements of different kinds: $\mathbf{X} \subseteq \mathbf{A}$, $\mathbf{Y} \subseteq \mathbf{B}$.

The peculiarities of the task depend on the text units considered: words (as in our examples) [12], clauses [10], sentences [1][4][6], paragraphs, sections, etc.; see Fig. 3.

If the units are too large, such as whole sections, the task is usually trivial: the text and its translation consist of the same number of sections, which correspond to each other in the natural order. On the other hand, if the text units are too small—such as morphemes or words—the very definition of the task becomes complicated, as Fig. 2 shows. In particular, in this case there are elements without translations, as in Fig. 1 (*a*), the order of the elements is not preserved, as in Fig. 1 (*b*), or even the indivisible groups of elements may be not contiguous, as in Fig. 2 (*c*).

The medium-size units such as sentences and paragraphs are an intermediate case: while the alignment task is not trivial, it does not usually present most of the complications discussed above. Particularly in the case of paragraphs, the order of the elements is preserved and every element has a translation, cf. Fig. 1. What is more, we assume that the translator can join two or more source paragraphs into one translated paragraph or split one source paragraph into several translated paragraphs, as in

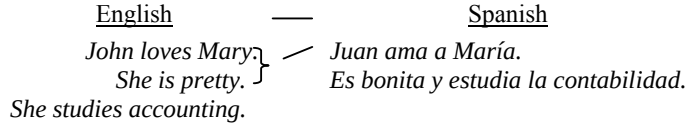

Fig. 3. Sentence-level alignment (literally: 'She is pretty and studies accounting').

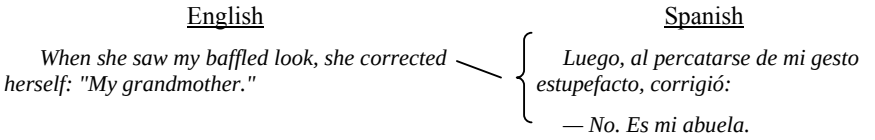

Fig. 4. One-to-many paragraph alignment (here the direct speech is translated as a separate paragraph).

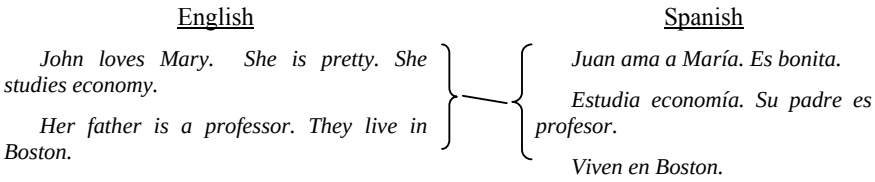

Fig. 5. An artificial example of many-to-many paragraph alignment (here the third and fourth sentences are translated as a separate paragraph).

Fig. 4, cf. Fig. 2 (*a*) and Fig. 3, but cannot re-arrange the sentences in the paragraphs in such a way that, say, a final part of a paragraph be translated as a beginning of another paragraph, as in Fig. 5, which would lead to patterns such as those shown in Fig. 2 (*b*, *c*). Though these assumptions are not completely true to life, they significantly simplify formalization of the task and the algorithm.

Thus, we define the paragraph-level bilingual text alignment task as the problem of constructing a bipartite hypergraph (cf. Fig. 2 (*c*)) whose vertices are the paragraphs of the texts in the two languages, respectively, and whose hyperarcs—standing for the sets of vertices to be mutual translations—satisfy the following conditions:

- Every vertex is incident to an arc, i.e., no paragraph disappears or appears from nothing in the translation process;
- At least one side of each arc has only one end, i.e., an arc can connect element to element, element to group, group to element, but not group to group;
- The ends of each arc are contiguous, i.e., a group of paragraphs that is the translation of a paragraph in the other language occupy a contiguous run of the text;
- The arcs are not crossing, i.e., the order of paragraphs is preserved in translation.

The (sets of) paragraphs that are translations of each other are *similar* in the sense of Section 3 below. This similarity measure can be assigned to the hyperarc connecting the paragraphs; we call this value the *weight* of the hyperarc. Our hypothesis is that the total weight of all hyperarcs gives the quality of a particular alignment. With this, the task is reduced to finding, among all possible hypergraphs satisfying the above conditions, the one with the maximum total weight of its hyperarcs.

3 Distance Measure

To assign the weight to a hyperarc as described in Section 2 above, we need to calculate the similarity between two sets of paragraphs (in our algorithm at least one of the

two sets consists of only one paragraph). We define it as the similarity between the two texts that are obtained by concatenation of the corresponding paragraphs.

3.1 Baseline Distance Measure

Common sense suggests that the corresponding pieces of texts are located at approximately the relative same distance from the beginning of the whole text. We define the baseline distance between two pieces of text, T_A in the language A and T_B in the language B, as follows:

$$\text{Distance}(T_A, T_B) = |\text{start}(T_A) - \text{start}(T_B)| + |\text{end}(T_A) - \text{end}(T_B)|, \quad (1)$$

where start(T_X) is the relative position of the first word of the text T_X measured in percentage of the total number of words in the text in the corresponding language, and similarly for end(T_X). We could also use the position of the paragraph instead of word as percentage of the total number of paragraphs, but the measure based on word counts has been reported as better than the one based on paragraph counts, which agrees with our own observations.

3.2 Proposed Distance Measure

We define the similarity between two texts in different languages as the number of words in both texts that are not mutual translations of each other [12]. For this, we first define which words are such translations.

1. Set T_X, := the shortest one of T_A and T_B; T_Y := the longest
2. Set translations := 0
3. for each word token w in T_X
4. if any of its translations $D_{XY}(w)$ is found in T_Y
5. increase translations by 1

where $D_{XY}(w)$ is a function returning a set of the dictionary translations of the word w. Then the number of word tokens without translation in both paragraphs, under the hypothesis that these two paragraphs correspond to each other, is:

$$\text{Distance}(T_A, T_B) = |T_A| + |T_B| - 2 \times \text{translations}. \quad (2)$$

The cost of an alignment hypothesis is the total number of words in both texts that are left without translation under this hypothesis. Note that under different hypotheses this number is different: here we consider two word tokens to be translations of each other if both of the following conditions hold: (a) they are dictionary translations (as word types) and (b) the paragraphs where they occur are supposed to be aligned.

Note that we represent the texts as vectors of word frequencies, ignoring the order of the words. In particular, concatenation of the paragraphs into text pieces is performed simply as summation of the corresponding vectors.

The above algorithm for calculating the number of translations has a drawback: in line 5, the same word in T_Y can potentially be counted twice, as in the English sentence T_A = "*The bank is at the French border*" and Spanish sentence T_B = "*Juan vive a*

la orilla de la ciudad" 'John lives at the border of the city.' However, addressing this problem would lead to a more complicated and computationally more expensive algorithm, which we may consider in our future work.

Recall that in our formalization of the task we select the optimal hypergraph out of hypergraphs with different number of arcs. This leads to that the algorithm would usually prefer a smaller number of hyperarcs: in an extreme case it might tend to align the firs paragraph of *A* with all but one paragraphs of *B*, and the rest of *A* with the last paragraph of *B*, which gives only two hyperarcs. However, the specific measure we suggest here is linear in the sense that it does not depend on the number of arcs: the cost is proportionally greater for the hyperarcs that align more paragraphs with one. Thus with this particular measure the algorithm is not biased towards a smaller number of larger pieces being aligned. Note that our experiments show (see Table 2) that our baseline measure suffers from such a bias.

4 Algorithm

To find the exact optimal alignment, we apply a dynamic programming algorithm. It uses a $(N_A + 1) \times (N_B + 1)$ chart shown in Fig. 6, where N_X is the number of paragraphs in the text in the language X.

The algorithm works as follows. First, the chart is filled in:

1. $a_{00} := 0$, $a_{i0} := -\infty$, $a_{0j} := -\infty$ for all $i, j > 0$.
2. for i from 1 to N_A do
3. for j from 1 to N_B do
4. $a_{ij} := \min (a_{xy} + \text{Distance} (T_A[x + 1 .. i], T_B[y + 1 .. j]))$

Here, a_{ij} is the value in the (i,j)-th cell of the chart, $T_X[a .. b]$ is the set of the paragraphs from a-th to b-th inclusive of the text in the language X, and the minimum is calculated over all cells (x,y) in the ⌟-shaped area to the left and above the (i,j)-th

Language *B*

Language *A*	0	1	2	*j*	...	N_B
0	0	∞	∞	∞	...	∞
1	∞	0.1	0.3	0.4	0.6	0.8
2	∞	0.3	0.5	0.5	0.7	0.7
3	∞	0.4	0.7	0.7	0.8	0.9
i	...	0.4	0.6	a_{ij}		
N_A	∞					?

Fig. 6. The chart of the dynamic programming algorithm.

cell, as marked with a triple-line border in Fig. 6. In our implementation we star from the corner of this area, thus preferring of equal variants the ones with fewer paragraphs being aligned with one paragraph. Note that at least one of the two $T_X[a \,..\, b]$ consists of only one paragraph, according to the conditions from Section 2. In Fig. 6, arbitrary values such as 0.3 are shown only to indicate that the corresponding cells have been already filled by the step of calculating a_{ij}.

As in any dynamic programming algorithm, the value a_{ij} is the total weight of the optimal alignment of the initial i paragraphs of the text in the language A with the initial j paragraphs of the text in the language B. Specifically, upon termination of the algorithm, the bottom-right cell (marked by "?" in Fig. 6) contains the total weight of the optimal alignment of the whole texts. The alignment itself is printed out by restoring the sequence of the assignments that led to this cell:

```
5.  (i,j) := (N_A, N_B).
6.  while (i,j) ≠ (0, 0) do
7.      (x,y) := argmin (a_xy + Similarity (T_A[x + 1 .. i], T_B[y + 1 .. j]))
8.      print "paragraphs in A from x + 1 to i are aligned with
9.              paragraphs in B from y + 1 to j."
10.     (i,j) := (x,y)
```

Here, again, the minimum is sought over the same ⌟-shaped area to the left and above the current cell (i,j). Upon termination, this algorithm will print (in the reverse order) all pairs of the sets of paragraphs in the optimal alignment. Note again that in each pair at least one of the two sets consists of only one paragraph. We omit here the proof of optimality, which is quite standard for dynamic programming algorithms.

5 Complexity Analysis

In this paper our goal was to prove that the task of finding the exact optimal alignment is tractable, and present the general idea of the algorithm. We did not have the goal of discussing its fast implementation.

The algorithm as presented here has the complexity $O(N^4)$, where $N = N_A \approx N_B$ is the size of the text to be aligned. Indeed, the chart contains $O(N^2)$ cells, calculating of each cell requires $O(N)$ calculations of similarity between one paragraph and a set of paragraphs, which in turn can require $O(N)$ comparisons of individual words. We assume that the size of a paragraph and the number of translations for a word in the dictionary are $O(1)$.

The algorithm can be trivially modified to have the complexity $O(N^{3.5})$. Indeed, the Heaps law [2] states that the number of different word types in a text of length N is $O(N^{0.5})$. Assuming a faster implementation of the algorithm from Section 3.2 dealing with word vectors and not with individual tokens, we get the complexity of the function Similarity (T_A, T_B) to be $O(N^{0.5})$ instead of $O(N)$. In case of our suggested distance measure, the complexity can be even lowered to $O(N^3)$ by incremental calculation of the distance in the inner loop of the algorithm, due to linearity of our measure.

In practice, the complexity can be lowered to $O(N^2)$ by limiting the size of the ⌟-shaped area in the chart calculation (which will also reduce to $O(1)$ the calculation of the similarity function). Indeed, as reported in [8], the correspondences longer that 1 to 3 paragraphs are low-probable.

We even believe that a linear on average (though not in the worst case) algorithm can be constructed, but this should be a topic of a future research.

6 Experimental Results

We experimented with a science fiction novel *Advances in genetics* by Abdón Ubídia and its original Spanish text *De la genética y sus logros*, downloaded from Internet. The English text consisted of 114 paragraphs and Spanish 107, including the title.[2] The texts were manually aligned at paragraph level to obtain the gold standard, see Table 1. In this table, only non-one-to-one pairs are shown: e.g., 2–3=2 stands for the fact that English paragraphs 2 and 3 constitute the translation of Spanish paragraph 2. The one-to-one pairs are trivially inferred from the data shown in the table: for example, 48–50=47 continues as 51=48, 52=49, etc.

Table 1. Comparison of the methods.

Method	Alignment
Gold	2–3=2; 4–5=3–4 (4–5=3, 5=3–4); 6=5–6; 9–10=9; ∅=21; 46–47=46; 48–50=47; 51–53=48; 58–59=53; 87–88=81
Proposed	2–3=2; 4–5=3; 6=4–6; 9–10=9; 22=21–22; 46–47=46; 48–50=47; 51–53=48; 58–59=53; 67=61–62; 68=63–64; 69–71=65; 85=79–80; 86–88=81
Baseline	2–3=2; 4–6=3; 7=4–7; 9–10=9; 11–12=10; 13=11–13; 15–16=15; 22=21–23; 23–24=24; 25–32=25; 33=26–27; 35=29–32; 36–37=33; 38=34–35; 39=36–38; 41=40–41; 42–43=42; 44=43–44; 46–47=46; 48–50=47; 51–53=48; 54=49–50; 55–56=51; 57–58=52; 59–60=53; 61=54–55; 63–64=57; 65=58–60; 66–68=61; 69=62–63; 72–73=66; 76–77=69; 78=70–71; 79=72–73; 82=76–77; 83–84=78; 85=79–80; 86–87=81; 88–89=82; 91–92=84; 94–96=86; 97=87–89; 98=90–91; 99–100=92; 101–102=93; 103–104=94; 105=95–99; 106–108=100; 109=101–102; 111–112=104; 113=105–106

As often happens with literary texts [15], the selected text proved to be a difficult case because of violation of two of our assumptions; see the underlined pairs in Table 1. In one case, two paragraphs were aligned with two: the translator broke down a long Spanish paragraph 3 into two English paragraphs 4 and 5, but joined the translation of a short Spanish paragraph 4 with the English paragraph 5; in Table 1 we illustrate this situation in parentheses. In another case, the translator completely omitted the Spanish paragraph 21. This illustrates that our assumptions from Section 2 above are not always correct. Obviously, our algorithm (with both distance measures) did not align correctly these cases.

[2] We did not experiment with a larger corpus because we are not aware of a gold-standard manually aligned Spanish-English parallel corpus.

Both texts were preprocessed by lemmatizing [7], [19] and POS-tagging, which allowed for correct dictionary lookup. Stop-words were removed to reduce noise in comparison; leaving the stop-words in place renders our method of comparison of paragraphs completely unusable. Then our algorithm was applied, with both baseline and suggested distance measures. The resulting alignments are shown in Table 1.

We evaluate the results in terms of precision and recall of retrieving the hyperarcs [10]; see Table 2: precision stands for the share of the pairs in the corresponding row of Table 1 (including one-to-one pairs not shown explicitly in the table) that are also found in its first row; recall stands for the stare of the pairs in its first row that are also found in the row corresponding to the method. Alternatively, we broke down each hyperarc into pair-wise correspondences: 48–50=47 was broken down into 48 ~ 47, 49 ~ 47, 50 ~ 47, and calculated the precision and recall of our algorithm on retrieving such pairs; see the last two columns of the table.

Table 2. Comparison of the distance measures.

Measure	Hyperarcs		Single arcs	
	Precision, %	Recall, %	Precision, %	Recall, %
Proposed	89	85	88	90
Baseline	65	28	43	54

One can see that the proposed distance measure based on the bilingual dictionaries greatly outperforms the pure statistically-based baseline.

7 Conclusions and Future Work

We have suggested a formalization of the paragraph alignment task as finding a least-cost hypergraph with certain properties. We also described a dynamic programming algorithm that finds the exact optimum of the corresponding problem. The assumptions that allowed for our formalization and thus the algorithm hold most of the time though not always, as our test corpus showed.

The following directions of future work can be mentioned:

- Error analysis: to analyze the causes of the errors made by the algorithm on our corpus. Actually only such analysis will define the ways of future improvements.
- Algorithm improved as to its complexity, preferably linear.
- More accurate similarity measures, for example, to avoid possibly counting some translations more than once. However, this is complicated and would lead to much higher complexity: in fact it implies word-level alignment.
- A similarity measure taking into account the order of words in the paragraphs.
- Weighting schemes such as TF-IDF instead of removing keywords.
- Formalization of the task considering many-to-many correspondences.
- Application of the method to sentence-level alignment.
- Using the results of alignment to enrich existing bilingual dictionaries.

References

[1] Brown, P. F., Lai, J. C. & Mercer, R. L. 1991. Aligning Sentences in Parallel Corpora. In: *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California, pp. 169–176.

[2] Baeza-Yates, R., B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999.

[3] Caseli, H. M., and M. G. Volpe Nunes. 2003. Evaluation of Sentence Alignment Methods on Portuguese-English Parallel Texts. *Scientia* 14(2):1–14.

[4] Chen, S. 1993. Aligning sentences in bilingual corpora using lexical information. In: *Proceeding of ACL-93*, pp. 9–16.

[5] Bing Zhao *et al*. 2003. Efficient Optimization for Bilingual Sentence Alignment based on Linear Regression. In: *HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*. p. 81–87.

[6] Gale, W. A. & Church, K. W. 1991. A program for Aligning Sentences in Bilingual Corpora. In: *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California.

[7] Gelbukh, Alexander, and Grigori Sidorov. 2003. *Approach to construction of automatic morphological analysis systems for inflective languages with little effort*. Lecture Notes in Computer Science, N 2588, Springer-Verlag, pp. 215–220.

[8] Gelbukh, Alexander, and Grigori Sidorov. 2006. Paragraph-Level Alignment of an English-Spanish Parallel Corpus of Fiction Texts using Bilingual Dictionaries. Proc. of TSD-2006. Lecture Notes in Artificial Intelligence, Springer-Verlag, in press.

[9] Kay, Martin and Martin Roscheisen. 1993. Text-translation alignment. *Computational Linguistics*, 19(1):121–142.

[10] Kit, Chunyu, Jonathan J. Webster, King Kui Sin, Haihua Pan, Heng Li. 2004. Clause alignment for Hong Kong legal texts: A lexical-based approach. *International Journal of Corpus Linguistics* 9:1. pp. 29–51.

[11] Langlais, Ph., M. Simard, J. Veronis. 1998. Methods and practical issues in evaluation alignment techniques. In: *Proceeding of Coling-ACL-98*.

[12] McEnery, A. M. & Oakes, M. P. 1996. Sentence and word alignment in the CRATER project. In: *Using Corpora for Language Research*, London, pp. 211–231.

[13] Melamed, I. Dan. 1996. A Geometric Approach to Mapping Bitext Correspondence. *Proc. EMNLP-1996*, ACL, p. 1–12.

[14] Melamed, I. Dan. 2000. Pattern Recognition for Mapping Bitext Correspondence. In *Parallel Text Processing: Alignment and Use of Translation Corpora*. Kluwer, p. 25–47.

[15] Meyers, Adam, Michiko Kosaka, and Ralph Grishman. 1998. A Multilingual Procedure for Dictionary-Based Sentence Alignment. In: *Proceedings of AMTA'98: Machine Translation and the Information Soup*, pages 187–198.

[16] Mikhailov, M. 2001. Two Approaches to Automated Text Aligning of Parallel Fiction Texts. *Across Languages and Cultures, 2:1*, pp. 87–96.

[17] Robert C. Moore. 2002, Fast and Accurate Sentence Alignment of Bilingual Corpora. *AMTA-2002*. p. 135–144.

[18] Simard, M., George Foster, Pierre Isabelle. 1992. Using Cognates to Align Sentences in Bilingual Corpora. TMI-1992, p. 67–81.

[19] Velásquez, F., Gelbukh, A. & Sidorov, G. 2002. AGME: un sistema de análisis y generación de la morfología del español. In: *Proc. of Workshop on Multilingual information access & natural language processing of IBERAMIA 2002*, pp 1–6.