View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Repositório da Universidade Nova de Lisboa

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Mestrado em Engenharia Informática

Parallel Texts Alignment

Luís Manuel dos Santos Gomes (aluno nº 26941)

1º Semestre de 2008/09
20 de Fevereiro de 2009

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Parallel Texts Alignment

Luís Manuel dos Santos Gomes (aluno nº 26941)

Orientador: Prof. Doutor José Gabriel Pereira Lopes

Trabalho apresentado no âmbito do Mestrado em Engenharia Informática, como requisito parcial para obtenção do grau de Mestre em Engenharia Informática.

1º Semestre de 2008/09
20 de Fevereiro de 2009

Agradecimentos

Em primeiro lugar quero agradecer ao meu orientador, o Professor Gabriel Pereira Lopes, pela confiança que depositou em mim, pelas orientações que me deu, pela paciência que teve e pelas palavras de encorajamento. Foi um enorme privilégio ter sido orientado pelo Professor Gabriel e esta experiência fez crescer o meu interesse pela investigação que já havia sido plantado em mim por *todos os Professores que me acompanharam* e me edificaram desde a minha primeira Professora, Maria de Jesus, a quem presto homenagem. A lista seria demasiado extensa e não conseguiria recordar os nomes de todos os Professores que tive, mas agradeço-lhes a dedicação e todas as horas de aulas, que na altura pareciam ser muito longas, e que hoje agradeço não terem sido mais curtas. Os Professores deram-me muito mais que instrução. Deram-me educação e alargaram-me os horizontes. Quem me ensinou, meu amigo foi(é). Não tendo sido meu professor, desempenhou um papel semelhante: Carl Sagan deu-me o Cosmos; hoje, não consigo olhar para um céu estrelado, ou pensar em Ciência, sem que me venha à memória a narrativa apaixonante e a clareza das suas exposições.

Agradeço ao Aires pela paciência que tem para me ouvir falar "daquela ideia que acabei de ter mesmo agora...". Agradeço à Filipa e ao Sérgio, sempre bem dispostos e sempre dispostos a tirar-me dúvidas de linguística.

Agradeço à Professora Maria João Romão e aos meus colegas do XTAL, com quem trabalhei vários anos e com quem tenho mantenho uma relação de amizade.

Agradeço aos meus tios, Aurora e João, que me acolheram na sua casa e me trataram como um filho. Infelizmente o meu tio João partiu no decurso desta dissertação. Agradeço à Paula Dinis Melo e ao Paulo Melo por serem uma inspiração para mim, e pela amizade entre nós. Ao Luís "grande-– em miúdo, eu era o Luís pequeno — com quem gosto imenso de conviver.

Agradeço aos meus Pais, a quem devo muito mais que a minha própria existência: se não tivessem dado o Amor e a dedicação que deram, se não me tivessem ensinado a respeitar os outros e mim próprio, se não me tivessem transmitido todo um conjunto de valores, eu não seria a mesma pessoa. A eles agradeço o Mundo que me deram e as portas que me abriram; Não poderiam ter sido melhores — e continuam a ser.

Agradeço aos Pais da Ana Luísa pela sua dádiva ao Mundo! Infelizmente o Pai da Ana Luísa partiu antes do tempo, e não tive o privilégio de o conhecer. Agradeço à D. Graciete e à Ana Luísa, os serões de convívio magníficos que me proporcionaram.

Agradeço à Ana Luísa por testemunhar a minha vida. Por tornar os dias mais azuis, as noites mais estreladas, os campos mais verdes, as cidades mais românticas, os filmes mais interessantes e as músicas mais bonitas. Agradeço-lhe a inspiração, a motivação e os conselhos que me dá. Agradeço-lhe pela perspectiva de um futuro bonito.

A todos agradeço por fazerem parte da minha vida.

Resumo

O alinhamento de textos paralelos (textos que são tradução um do outro) é um passo necessário para várias aplicações que utilizam esses textos, como é o caso da tradução automática estatística, a extracção automática de equivalentes de tradução, a criação automática de *concordances*, entre outras.

Nesta dissertação é apresentada uma metodologia para alinhamento de textos paralelos que, relativamente ao estado da arte, introduz mudanças importantes, tanto ao nível dos objectivos, como da forma como são concretizados. Uma grande parte dos métodos existentes tenta obter um léxico bilingue durante o processo de alinhamento que é usado para estabelecer correspondências entre os textos. Nalguns casos, esse léxico é complementado com um léxico pré-existente. Nesta dissertação são apresentadas várias razões que sustentam a tese de que o método de alinhamento *não deve incluir qualquer tipo de extracção automática de léxico*. Por conseguinte, a metodologia apresentada assenta *exclusivamente* em léxicos obtidos externamente, sendo apresentada uma solução técnica que permite o uso de um léxico extremamente grande.

Além da utilização exclusiva de um léxico externo, é apresentado um método inovador para obtenção de correspondências entre ocorrências de equivalentes de tradução nos textos. Esse método usa um critério de decisão baseado em propriedades das ocorrências que ainda não haviam sido exploradas por outros métodos.

O método é iterativo e converge para um alinhamento mais fino e mais correcto. À medida que o alinhamento é refinado, o método tira partido da nova informação para evitar correspondências erradas que haviam sido obtidas em iterações anteriores.

Palavras-chave: alinhamento de textos paralelos, corpora paralelos, extracção de equivalentes de tradução

Abstract

Alignment of parallel texts (texts that are translation of each other) is a required step for many applications that use parallel texts, including statistical machine translation, automatic extraction of translation equivalents, automatic creation of concordances, etc.

This dissertation presents a new methodology for parallel texts alignment that departs from previous work in several ways. One important departure is a shift of goals concerning the use of lexicons for obtaining correspondences between the texts. Previous methods try to infer a bilingual lexicon as part of the alignment process and use it to obtain correspondences between the texts. Some of those methods can use external lexicons to complement the inferred one, but they tend to consider them as secondary. This dissertation presents several arguments supporting the thesis that *lexicon inference should not be embedded in the alignment process*. The method described complies with this statement and relies exclusively on externally managed lexicons to obtain correspondences. Moreover, the algorithms presented can handle very large lexicons containing terms of arbitrary length.

Besides the exclusive use of external lexicons, this dissertation presents a new method for obtaining correspondences between translation equivalents found in the texts. It uses a decision criteria based on features that have been overlooked by prior work.

The proposed method is iterative and refines the alignment at each iteration. It uses the alignment obtained in one iteration as a guide to obtaining new correspondences in the next iteration, which in turn are used to compute a finer alignment. This iterative scheme allows the method to correct correspondence errors from previous iterations in face of new information.

Keywords: parallel texts alignment, parallel corpora, extraction of translation equivalents

Contents

List of Figures

List of Tables

1. Introduction

Two texts are deemed parallel if one is a translation of the other. It is also possible that both were translated from a common source text. Large quantities of parallel corpora[1] are available today from several sources, notably from the European Union (EU) resulting from it's strong multilingualism policy.

The JRC-Acquis corpus [38] is a large part of the *Acquis Communautaire*, the body of documentation with legal nature produced by the various institutions of the European Union, including Treaties, declarations and resolutions, legislation and international agreements. This large corpus of about one billion words contains texts in 22 languages making it the most multilingual corpus available today. Figure 1.1 presents an excerpt, in both English and Portuguese, from the Official Journal of the European Union that is part of the JRC-Acquis. Another large parallel corpus is the Europarl corpus [24] that comprises transcriptions from the Proceedings of the European Parliament.

Aligning two parallel texts consists of dividing both texts into a number of segments[2] such that the *i*th segment of one text corresponds to the *i*th segment of the other. Crossed correspondences between segments are not permitted (thus the name *alignment*) and this restriction is called the *monotonicity constraint*. An example alignment is presented in figure 1.2 — note that, for the Portuguese text, we separate contractions such as "da" into their constituents: preposition "de" and article "a".

The word *corresponds* is used with the meaning that one segment has been translated by the other despite the fact that such translation may not be exact if taken out of that context. For example in figure 1.2 the English word "being" (present continuous tense) was translated as "são" (present tense) in Portuguese. The exact translation of "being" is "sendo" but the translator opted to change the verb tense for stylistic reasons (the present continuous tense is not used as frequently in Portuguese as it is in English). Despite that change, the Portuguese sentence is well formed and conveys the same meaning of the English sentence, thus the translation is good.

A correspondence between two parallel texts is a mapping between segments of both texts. Crossed correspondences are possible as well as *discontiguous correspondences*, which are correspondences between one segment or several discontiguous segments in one text and several discontiguous segments in the other text. Figure 1.3 shows an example correspondence between the same texts on figure 1.2 and figure 1.4 presents more examples of discontiguous correspondences.

If there are regions in the texts not covered by the correspondence, then the correspondence is *partial*, otherwise it is *total*. The alignment in 1.2 is a total correspondence while the correspondence in figure 1.3 is partial because there are three Portuguese prepositions "de" that do

[1] A *corpus* is a collection of texts in machine-readable format; *corpora* is the plural of *corpus*.

[2] a segment is simply a part of the text; however, the word segment is preferred because it suggests a partition of the text along natural boundaries like word or sentence boundaries

. . .

On 12 May 2004 the Commission received notification, pursuant to Article 3(1)(b) of the Council Regulation, of a proposed merger by which Continental AG wished to acquire sole control over Phoenix AG, both undertakings being leaders on the rubber products manufacturing market.

Having examined the information submitted by the parties to the proposed merger and conducted a market survey, the Commission concluded that the merger raised serious doubts as to compatibility with the common market and the EEA Agreement.

. . .

(a) English version

. . .

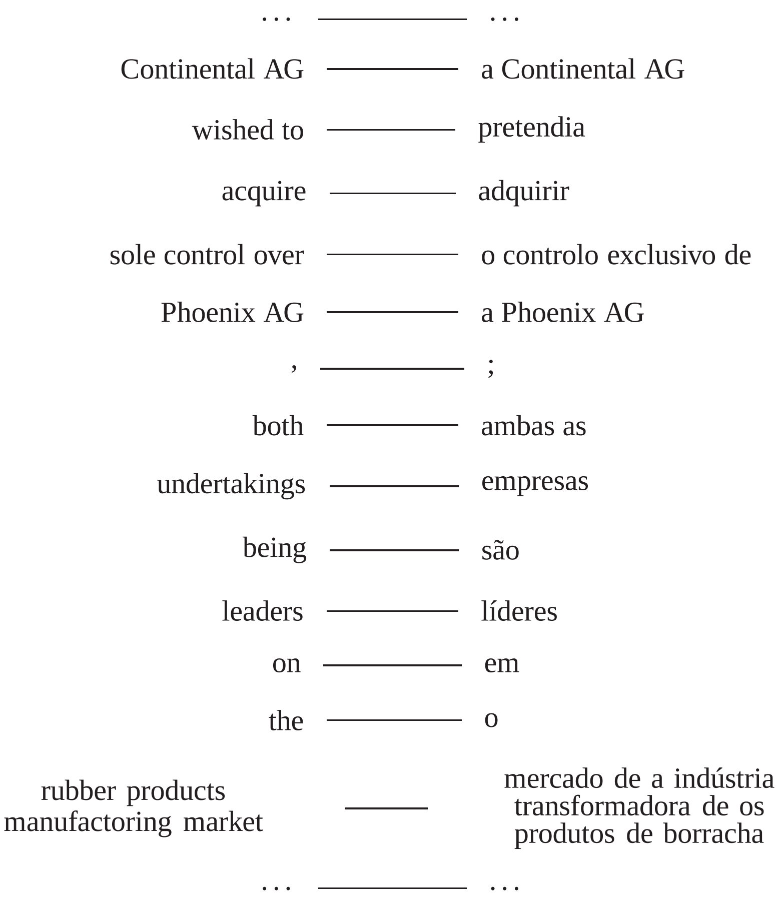
Em 12 de Maio de 2004, a Comissão recebeu uma notificação, em os termos de o n.o 1, alínea b), de o artigo 3.o de o Regulamento de o Conselho, de um projecto de concentração através de a qual a Continental AG pretendia adquirir o controlo exclusivo de a Phoenix AG; ambas as empresas são líderes em o mercado de a indústria transformadora de os produtos de borracha.

Após analisar as informações apresentadas por as partes em a concentração projectada e após realizar um estudo de mercado, a Comissão concluiu que a concentração suscitava sérias dúvidas quanto a a sua compatibilidade com o mercado comum e o Acordo EEE.

. . .

(b) Portuguese version

Figure 1.1: English (a) and Portuguese (b) versions of an excerpt from the Official Journal of the European Union, part of the JRC-Acquis corpus. As one can see from the example (if one understands Portuguese), the translations in this corpus are very tight and thus very appropriate for a range of automatic methods like extraction of translation equivalents, statistical machine translation or example-based machine translation.



. . .	. . .
Continental AG	a Continental AG
wished to	pretendia
acquire	adquirir
sole control over	o controlo exclusivo de
Phoenix AG	a Phoenix AG
,	;
both	ambas as
undertakings	empresas
being	são
leaders	líderes
on	em
the	o
rubber products manufactoring market	mercado de a indústria transformadora de os produtos de borracha
. . .	. . .

Figure 1.2: Alignment between English and Portuguese excerpts from the Official Journal of the European Union. Each segment is connected to it's translation with a line. Note that there are no crossed lines and that there is no text left unconnected. Note also that the last segment is large because words have a different order in both languages.

not correspond to words in the English side.

The term *alignment* "has become something of a misnomer in computational linguistics" (Wu [44]) because it has been used to denote either *alignment* or *correspondence* depending on the author. In this dissertation we use the term *alignment* with its proper meaning and we use *correspondence* to denote what other authors call a *non monotonic alignment*.

Figures 1.2, 1.3 and 1.4 present a sub-sentence-level correspondence but coarser granularities are possible, including chapters, sections, paragraphs and sentences. In general, the applications that use aligned parallel corpora benefit from more fine-grained correspondences. Research has been focused on sentence-level and sub-sentence-level alignment, which are hereafter simply designated as sentence alignment and sub-sentence alignment, respectively.

1.1 Importance of parallel texts alignment

> Existing translations contain more solutions to more translation problems than any other existing resource — Pierre Isabelle [20]

The insightful statement above captures the importance of identifying correspondences between parts of translated texts. And it regards the usefulness of using such parallel corpora for enabling machines to extract/mine a huge number of translation solutions.

Until the end of the 1980s, research on machine translation had been focused on rule-based approaches. The idea of using statistical techniques in machine translation was revived in 1988 by Brown et al [2]. They describe a statistical method that uses a large collection of translated sentences[3] obtained from parallel corpora. Two corpus-based approaches have since then gained more attention from researchers: statistical machine translation (SMT) and example based machine translation (EBMT) — according to John Hutchins [18] these are "the main innovations since 1990" in the field of machine translation.

Also in the early 1990s, Klavans and Tzoukermann [23] suggested that parallel corpora could be used for bilingual lexicography. They pointed out that dictionaries can be enhanced with statistical data from parallel corpora and, conversely, statistical methods applied to parallel corpora can benefit from data available in dictionaries.

The rising interest on parallel corpora triggered the research on parallel texts alignment, as it is a preliminary step required for most corpora-based applications.

[3] more precisely, a collection of pairs of sentences that are mutual translations

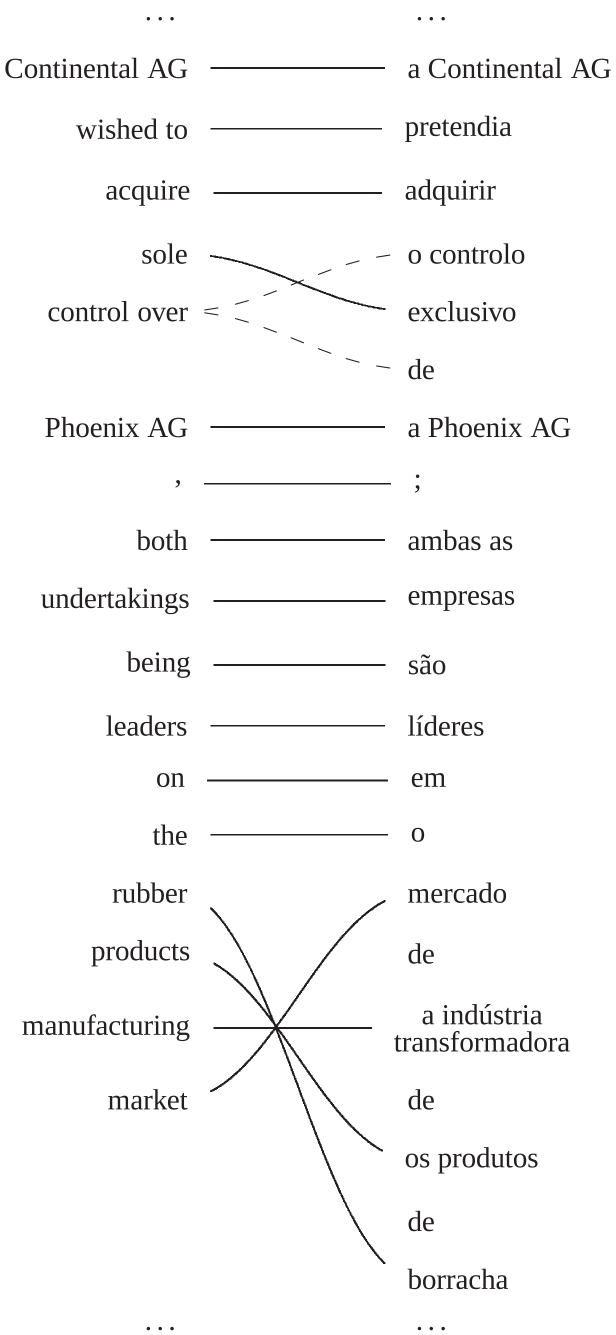


Figure 1.3: Correspondence between the same passages of figure 1.2. Discontiguous correspondences are represented with dashed lines. Compare this with the alignment in figure 1.2 and note how the text with crossed correspondences must be fitted into a single larger segment to comply with the monotonicity constraint of alignment.

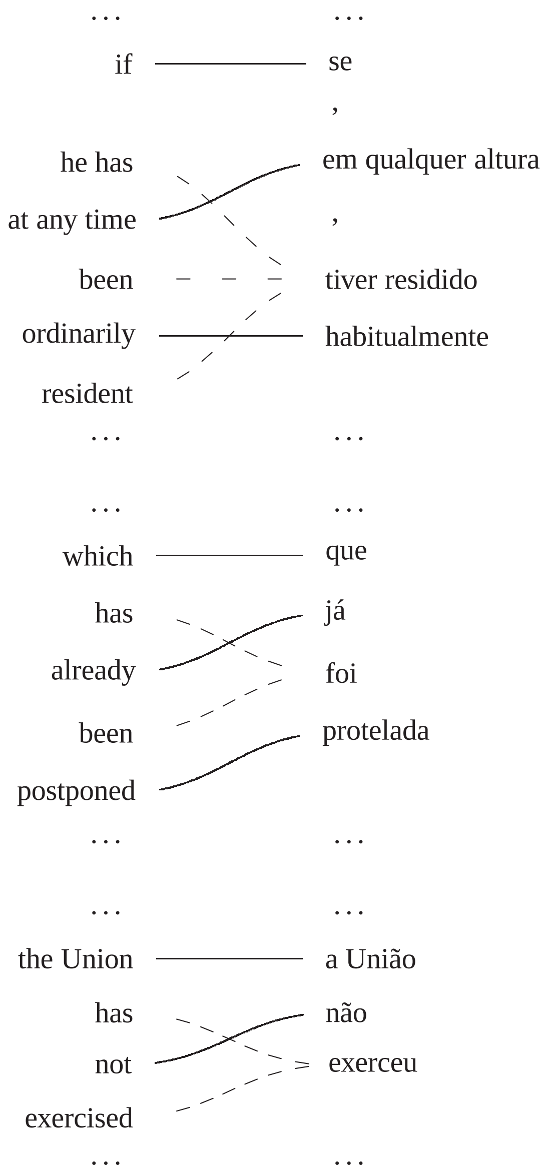


Figure 1.4: Selected examples of discontiguous correspondences (represented as dashed lines) between English expressions that use the auxiliary verb *has* and the respective Portuguese translations.

1.2 Objectives

Many bilingual dictionaries are available today, commercially and freely, in machine-readable format. For example, the website `http://freedict.org/` makes freely available[4] 64 dictionaries with over a million total headwords.

On the other hand, there are several methods (Kupiec [25], Dagan and Church [8], Wu and Xia [45], Wu [43], Melamed [27], Ribeiro et al [36, 35] and Ribeiro [31]) that can be used for extraction of single- and multi-word translation equivalents from aligned parallel corpora. These methods can be employed to build bilingual dictionaries, by performing hand verification of the automatically extracted translation equivalents.

One objective being sought is to take advantage of bilingual dictionaries to find correspondences in parallel texts.

Most of the previous work relies on linguistic concepts like *sentence*, *word*, *cognate*[5], *part of speech* or *grammar*. Each of the concepts has some degree of language dependence. Part of speech tagging is language dependent; grammars are even more; cognate words are frequent between close languages like Portuguese and Spanish and rare between unrelated languages like English and Japanese; Chinese and Japanese words are not delimited by whitespace like words in Indo-European languages, therefore, any method that relies on the concept of *word* must perform word segmentation for those languages. To summarize, methods that rely on any of those linguistic concepts will perform better or worse, depending on the pair of languages of the texts being aligned.

Another objective being sought in this work is to be as general as possible with regard to the languages of the texts being aligned. It should perform equally well for close and unrelated languages.

1.3 Main contributions

This work adds several important contributions to the state of the art:

1. Correspondences are obtained using a bilingual lexicon exclusively. This is a departure from previous methods, which try to infer a lexicon along with the alignment. An algorithm that uses suffix arrays with LCP information to locate multi-word translation equivalents in a text very efficiently. The algorithm is designed to cope with very large lexicons of terms with arbitrary length. This algorithm is presented in section 3.3.

2. A new method for obtaining correspondences that effectively avoids most of the noise

[4] under the GNU General Public License

[5] words of different languages that have derived from a common word, for example "president" in English and "presidente" in Portuguese, both derive from Latin "praesident"

from the search space, thus avoiding the need for filtering. The number of correspondences provided by this method depends almost exclusively on the size of the lexicon, meaning that, with a large lexicon it correctly identifies most correspondences in the texts. This method is explained in section 3.4.

3. An algorithm to obtain the "best" alignment from a set of correspondences. This algorithm, as well as the criteria to selecting the "best" alignment are explained in section 3.5.

1.4 Organization of this document

The four chapters of this dissertation provide a natural flow of reading, with each chapter preparing the reader for the next, thus I recommend a linear reading. I have strived for clarity and concision instead of verbosity.

Chapter 2 presents a perspective of the different approaches to the problem, how they relate to each other and how they compare. At the end of this chapter the reader will have a notion of the state of the art on parallel texts alignment techniques.

Chapter 3 describes the work done that suggests a new approach to parallel texts alignment.

Chapter 4 concludes this dissertation with a summary of the contributions followed by a list of topics suggested for further research. Because the contributed method innovates in many ways, the possibilities for further exploration are manifold.

2 . Previous Work

This chapter presents an overview of previous work on parallel texts alignment. Each section corresponds roughly to a family of methods that explore the same features of parallel texts. The new approach described in the next chapter relates more to the method described in the last section, which justifies the more detailed description of that method compared to the others.

2.1 Sentence alignment

The order of paragraphs is usually maintained in translation, albeit deletions or insertions of text may occur. Also, cases of one paragraph being translated as two paragraphs are rare, but not impossible. In more literal translations, like those in legal, medical and technical contexts, the order of sentences is also preserved for a large percentage: Gale and Church [13, 14] measured a one to one correspondence (without crossings) in 89% of the sentences of a corpus from the Union Bank of Switzerland.

The early research on parallel texts alignment focused on sentence alignment but sub-sentence alignment has gathered more interest from researchers as can be confirmed by the number of publications on each subject matter.

Sentence alignment requires sentence boundary detection which is usually done resorting to heuristics with variable accuracy depending on the corpus.

2.1.1 Length-based sentence alignment

The first sentence alignment methods (Gale et al [13, 16] and Brown et al [3]) used the length of sentences as the only feature of the text to induce the most probable alignment between sentences of two parallel texts. This appears to be a very shallow feature but the method performs well for clean texts (as opposed to noisy texts which are discussed in section 2.2) with fairly literal translations like the Canadian Hansards or the corpus of the Union Bank of Switzerland. Later methods (Simard et al [37], Wu [42]) combine this length-based approach with lexical information.

The main hypothesis used for length-based alignment is that parallel texts (including chapters, sections, paragraphs, sentences, whatever) tend to have proportional lengths measured either in terms of number of characters (Gale and Church, [13, 14]) or in terms of number of words (Brown et al [3]). Figure 2.1 shows that the lengths of a text and of it's translation are highly correlated.

Both teams considered six possible translation configurations: 1:1 (one sentence being translated by exactly one sentence), 0:1 (sentence inserted in the translated text), 1:0 (sentence not

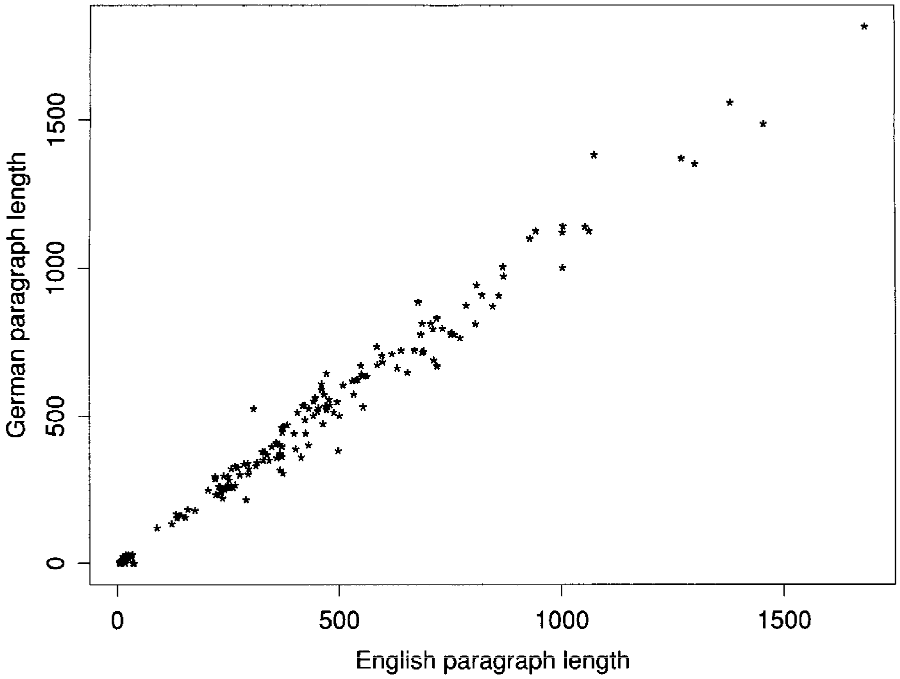


Figure 2.1: High correlation (0.991) between the lengths of mutual translations. The horizontal axis shows the length (measured in characters) of English paragraphs, while the vertical scale shows the lengths of the corresponding German paragraphs. Reproduced from [14].

translated), 1:2 (sentence translated as two sentences), 2:1 (two sentences translated as one sentence) and 2:2 (two sentences are translated by two sentences). While other translation configurations are possible, they are very rare, so both methods don't consider them. The most common configuration is 1:1, accounting for roughly 90% of the cases (Brown et al [3]).

Both methods approach the alignment problem as a maximum-likelihood estimation problem, though they use different probabilistic models.

The algorithm tries to find the most probable text alignment by using a dynamic programming algorithm which tries to find the minimum possible distance between the two parallel texts. Let $\mathrm{D}(i,j)$ be the lowest distance alignment between the first i sentences in text X and the first j sentences in text Y. Let $\ell_{x:i}$ denote the length of the ith sentence from X and $\ell_{y:j}$ denote the length of the jth sentence from Y.

Using $\mathrm{D}(0,0)=0$ as the base case, $\mathrm{D}(i,j)$ can be recursively defined as

$$D(i,j)=\min\begin{cases}\mathrm{D}(i,j-1)-\log\mathrm{P}(\alpha_{0:1}|\delta(0,\ell_{y:j}))\\ \mathrm{D}(i-1,j)-\log\mathrm{P}(\alpha_{1:0}|\delta(\ell_{x:i},0))\\ \mathrm{D}(i-1,j-1)-\log\mathrm{P}(\alpha_{1:1}|\delta(\ell_{x:i},\ell_{y:j}))\\ \mathrm{D}(i-1,j-2)-\log\mathrm{P}(\alpha_{1:2}|\delta(\ell_{x:i},\ell_{y:j}+\ell_{y:j-1}))\\ \mathrm{D}(i-2,j-1)-\log\mathrm{P}(\alpha_{2:1}|\delta(\ell_{x:i}+\ell_{x:i-1},\ell_{y:j}))\\ \mathrm{D}(i-2,j-2)-\log\mathrm{P}(\alpha_{2:2}|\delta(\ell_{x:i}+\ell_{x:i-1},\ell_{y:j}+\ell_{y:j-1}))\end{cases}$$

The term $\mathrm{P}(\alpha_{a:b}|\delta(\ell_x,\ell_y))$ is the probability of alignment configuration $\alpha_{a:b}$ given the lengths ℓ_x and ℓ_y of the portions of text under consideration. That probability is passed to the log domain so that it can be regarded as a distance (smaller probabilities correspond to greater distances).

According to Baye's law $\mathrm{P}(\alpha_{a:b}|\delta(\ell_x,\ell_y))$ is calculated as

$$\mathrm{P}(\alpha_{a:b}|\delta(\ell_x,\ell_y))=\frac{\mathrm{P}(\alpha_{a:b})\,\mathrm{P}(\delta(\ell_x,\ell_y)|\alpha_{a:b})}{\mathrm{P}(\delta(\ell_x,\ell_y))}$$

They simplify this equation by dropping the denominator and using an approximation

$$\mathrm{P}(\alpha_{a:b}|\delta(\ell_x,\ell_y))\approx\mathrm{P}(\alpha_{a:b})\,\mathrm{P}(\delta(\ell_x,\ell_y)|\alpha_{a:b})$$

Gale and Church [13, 14] approximate the conditional probability $\mathrm{P}(\delta(\ell_x,\ell_y)|\alpha)$ by

$$\delta(\ell_x,\ell_y)=\frac{\ell_y-\ell_x\mu}{\sqrt{\ell_x\sigma^2}}$$

where ℓ_x and ℓ_y are measured as the number of characters of proposed aligning sentences. Parameters μ (mean) and σ^2 (variance) are learned from the same training parallel corpus hand aligned which is also used to learn the apriori probability $P(\alpha)$ for each alignment configuration.

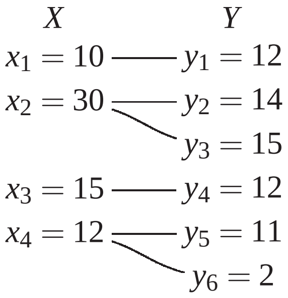
Brown et al [3] approximate the conditional probability $P(\delta(\ell_x,\ell_y)|\alpha)$ by

$$\delta(\ell_x,\ell_y) = log\frac{\ell_y}{\ell_x}$$

where ℓ_x and ℓ_y are measured as the number of words in proposed aligning sentences.

The apriori probability of alignment configuration α is higher, when hypothesized translation configuration is 1:1. The values for these probabilities are determined from a training parallel corpus hand aligned.

For example, given a representation of the two parallel texts as a sequence of sentence lengths, as bellow (adapted from Wu [44]) the most probable alignment is depicted as lines connecting sentences of X and Y:



Evaluation carried out by Gale and Church [14] show that using characters instead of words, maintaining all factors constant, yields higher accuracy (in their experiments, they have obtained a error rate of 4.2% for characters compared with 6.5% for words).

2.1.2 Lexical sentence alignment

Length-based alignment methods provide good results for rather well behaved parallel texts such as those from the Canadian Hansard and the Union Bank of Switzerland. However, they rely too much on the correct identification of sentence boundaries and they make no use of lexical information.

Simard et al [37] add lexical information to the method of Gale and Church [13] using possible cognates as lexical cues. They use a (rather shallow) heuristic for cognate detection that considers as cognates pairs of words that share a common prefix of 4 characters. They report that their method improves the original length-based method of Gale and Church [13], reducing the error rate by 10% at the expense of increasing the computation time by 12%.

Gale and Church [15] start with a sentence alignment using the method described in [13] and then use a word association measure to select word pairs with higher association score.

That score is computed from contingency tables calculated over a corpus. The tables contain the number of times that a pair of words co-occurs in the same segments (using the existing sentence alignment), and the total number of occurrences of each word. The association measure is higher for pairs of words that tend to co-occur. Because it is not computationally feasible to compute the association measure for all word pairs in a large corpus they use a "progressive deepening strategy" to select word pairs. Then, they try to match word pairs within sentences using a monotonicity constraint: match the first occurrence in one sentence with the first occurrence in the aligned sentence, the second with the second and so on. They report that this method gives a correspondence for 60% of the words that is correct 95% of the cases in their experiment.

Kay and Röscheisen [21, 22] describe an iterative algorithm that mutually induces a sentence alignment and a word correspondence map. The word correspondence is used to compute a maximum likelihood sentence alignment and the sentence alignment is used to refine the word correspondence map. Like Gale and Church [15], their method uses an association measure to induce word correspondence. They resort to heuristics to restrict the word correspondence search space that would otherwise be quadratic with respect to the length of the texts. The steps of their method are basically the following:

1. Assume that first and last sentences of text X align with the first and last sentences of text Y, respectively. These are the initial *anchors*.

2. Then, until most sentences are aligned:

   (a) Form an envelope of possible alignments from the Cartesian product of the list of sentences in both texts, not far from the diagonal line that crosses both anchors and not far from one of the anchors. The envelop is pillow shaped as shown in figure 2.2.

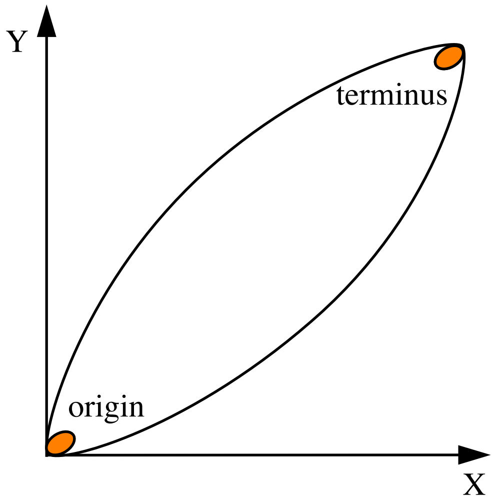
   (b) Choose pairs of words that tend to co-occur in potential partial alignments and whose similarity is higher than a given threshold. For this purpose, they use a known association measure (Dice coefficient, mutual Information, SCP, cosine).

   (c) Find pairs of source and target sentences which contain many possible lexical correspondences. The most reliable of these pairs are used to induce a set of partial alignments which will be part of the final result. Commit to these alignments adding them to the list of anchors, and repeat the steps above.

The alignment precision of this method applied on Canadian Hansard was measured marginally higher than length based alignment methods (Chen [4]).

We can see a common pattern in lexical alignment methods: they simultaneously induce a word correspondence and a sentence alignment that maximize the likelihood of each other. As a consequence, these methods build a bilingual lexicon as a by-product. For finding out similarity

Figure 2.2: Pillow shaped envelope of possible alignments.



of a word w_x in one language and its possible translation, w_y, these methods use association measures. Kay and Rösenschein used Dice Coefficient $(2\frac{f(w_x,w_y)}{f(w_x)+f(w_y)}$, where $f(w_x,w_y)$ denotes co-occurrence frequency of word w_x and its possible translation w_y, $f(w_x)$ denotes the frequency of w_x, and $f(w_y)$ the frequency of w_y. Haruno and Yamazaki [17] used Mutual Information $\log N\frac{f(w_x,w_y)}{f(w_x)*f(w_y)}$, N being the total number of segments.

2.2 Robust techniques

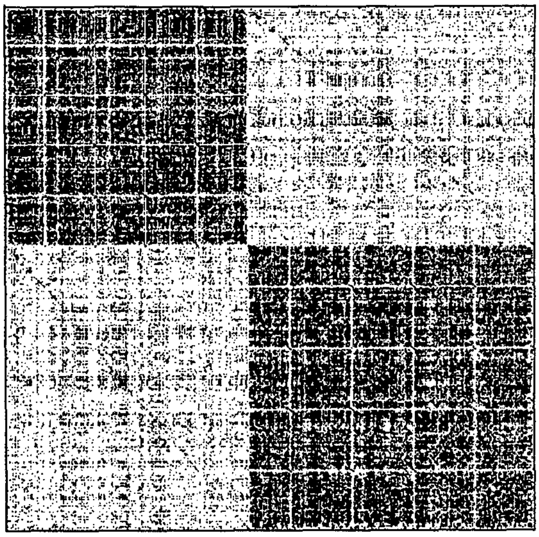
Robust methods are a class of methods targeted at "noisy" texts. They are more robust in face of non-literal translations, reordering of sentences or paragraphs, omissions, floating materials like footnotes, figures, headers, etc. Unlike the methods mentioned in the previous sections, robust methods do not require sentence boundary detection.

The output of robust methods is a set of points (x,y) that indicate correspondence between offset x and y in the respective texts.

2.2.1 Dotplots and signal processing

The first robust method, suggested by Church [6], was based on dotplots. A dotplot is a scatter plot as shown in figure 2.3. A dot is placed at (x,y) if there is a common character n-gram (they used 4-grams) beginning at position x and at position y of the input text. They concatenate the two texts and use the resulting text as the input for the dotplot. The top left and the bottom right quadrants of figure 2.3 have more points because each text is similar to itself than to the other (remember that the texts are concatenated). The useful information resides in

Figure 2.3: Example dotplot of two parallel texts concatenated. A point is plotted at (x, y) if the 4-gram starting at position x is the same as the 4-gram beginning at position y of the input (origin is at top left). The top left and bottom right quadrants have more points because each text is more similar to itself than to the other. Reproduced from [6].



the other two quadrants, because they indicate matches between the parallel texts. Figure 2.4 presents the upper right quadrant after being enhanced with signal processing techniques. The correspondences follow a diagonal line that comes from the fact that, despite local reordering of words or sentences, the order in which information is presented in a text is preserved in translation. This diagonal has been dubbed the *golden diagonal*.

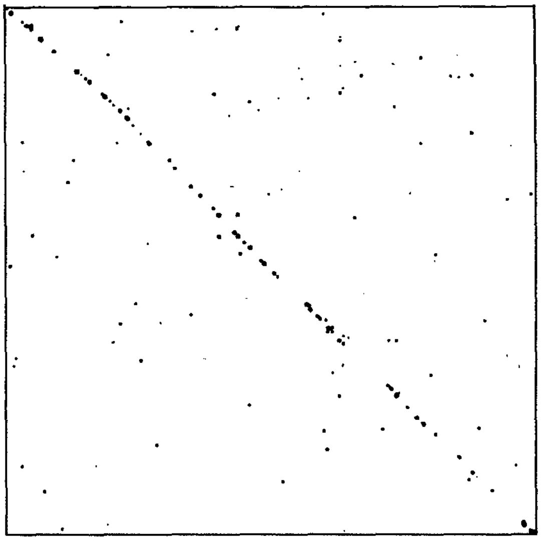
2.2.2 Methods based on regression lines

At UNL, Ribeiro et al [33, 34, 32] and Ribeiro [31] approach the alignment problem as a global restriction based on the hypothesis that correspondence points should be near the golden diagonal. First they obtain a set of candidate points using a simple frequency based heuristic and then they filter out noisy points using statistical methods. The method is applied recursively to each segment bounded by consecutive points.

In the first experiment [33], they used homograph tokens (numbers, proper names, punctuation signs, whatever) as lexical cues. To obtain correspondence points they select homograph tokens having identical frequency in both texts. This allows a simple pairing of the occurrences: the ith occurrence of a token t in one text is paired with the ith occurrence of t in the other text.

Using the offsets of the paired occurrences as possible correspondence points, they define a straight line that best fits the set of hypothetical correspondence points. Then, noisy points

Figure 2.4: Detail of the upper right quadrant of the dotplot in figure 2.3 enhanced by signal processing techniques. The correspondences line up along the *golden diagonal*. Reproduced from [6].



should be sifted out by "using statistically defined filters based on linear regression lines, rather then resorting to heuristics" — Ribeiro [31].

In figure 2.5, the arrow is pointing to a clearly noisy point corresponding to the single occurrence of word "integral" in those parallel texts, in rather different contexts. The graph makes it clear that that word is far away from its expected position on the regression line, whose equation is shown at the top right corner of the graph. The dotted vertical line enables one to see how far it is from its expected position.

The next step in the method is to apply two statistically supported filters based on the regression line.

The first filter is based on the histogram of distances between the original and the expected position of each candidate to be an aligner point. This is a coarse filter that aims at identifying extreme points (outliers), which are clearly far apart from their expected positions, preventing them from being considered reliable correspondence points. Figure 2.6 shows the effect of applying the first filter to the set of points in figure 2.5. The regression line is recomputed after the application of the first filter.

The second filter is based on confidence bands of linear regression lines (Wannacott and Wannacott [41] p. 334). It is a very fine grained filter; figure 2.7 shows the effect of applying the confidence-bands-based filter to the set of points of figure 2.6.

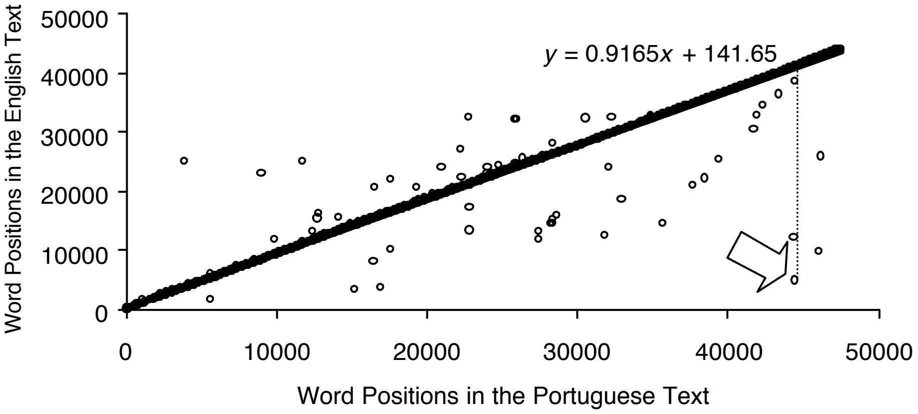


Figure 2.5: Noisy points versus "well behaved" points ("in line"). Reproduced from [31].

After applying both filters the method is applied recursively to each segment of the texts between two correspondence points. There are homographs that were not considered previously because their frequency was different for the whole parallel texts. However, in some of the smaller segments that were obtained meanwhile, those homographs may have the same number of occurrences. Moreover, as at a global level, the confidence band filter tend to be very selective — just ¼ of candidate points are retained as reliable — by looking at the parallel text segment level, where some of the previously discarded potential correspondence points may occur with identical frequency, those potential correspondence points will pass again by the same kind of procedure (linear regression, application of histogram filter, and application of the confidence band filter, at a local context) and locally they may be confirmed or refuted.

In a subsequent experiment (described in Ribeiro et al [32]), as European languages have a huge number of cognates (words having similar forms and meaning the same, as is the case for "Constitution" and "Constituição") and homographs are just cases of possible cognates, the number of candidate correspondence points were considerably enlarged by taking into account possible cognates with identical frequency. Existing proposals for determining possible cognates in two parallel texts (Simard et al [37]; Melamed [28] Danielsson et al, 2000), were rather unappealing. So, Ribeiro, Gael, Lopes and Mexia [32], decided to join the two texts to be aligned and extract from there relevant sequences of characters eventually having gaps, as would be the case for the character sequence "#_overn" (where the cardinal character "#" replaces the blank space and the underline replaces any character) which is common to the sequence of characters "#Government" and "#governo". For this purpose, the technique applied for extracting these sequences was the one which gave rise to Gael's Ph.D. Thesis [10] and Silva et al [7].

While earlier work was done at the level of words or tokens, where just the offset of the word/token starting character is worth considering, in this experiment, it was necessary to take

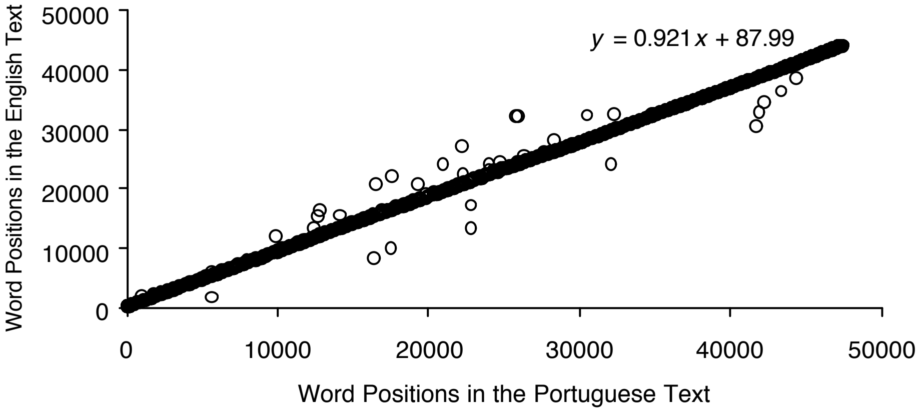


Figure 2.6: New set of candidate correspondence points after the application of the histogram of distances filter. At the right top is the new regression line equation. Reproduced from [31].

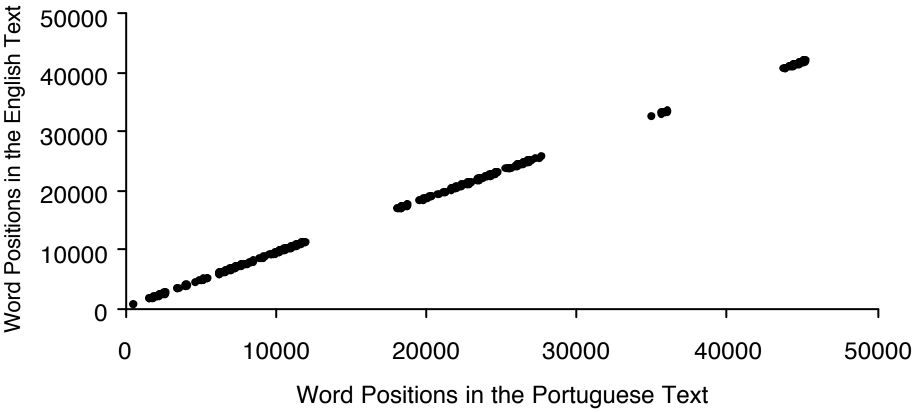


Figure 2.7: Selected correspondence points after the histogram filter and the confidence band filter have been applied. Reproduced from [31].

into account the position of the initial character of the character sequence and its length. And the same alignment technique was applied:

1. Identical character sequences with or without gaps, equally frequent, were selected as possible candidates for alignments. Selected sequences should not overlap.
2. Linear regression was applied.
3. Same kind of filters (histogram filter and the confidence band filter) were applied
4. For each aligned character sub-segment, repeat steps 1 to 4.

Ildefonso and Lopes [19] observed that the confidence bands based filter is computationally demanding and sifts out a huge number of candidate correspondence points which are good anchors, replaced that filter by a filter that selects longest sorted sequence of points. He opted to use the edit distance (Levenstein distance) to determine possible candidate cognates instead of the more sophisticated method proposed by Ribeiro et al [32].

2.3 Summary

We may identify two main trends for aligning parallel texts.

The first one, described in section 2.1, is targeted at sentence alignment and mixes several heuristics and statistics. A common pattern in those methods is to simultaneously induce a word correspondence and a sentence alignment that maximize the likelihood of each other.

The second one, described in 2.2, acknowledges the need for robustness when dealing with "real world" texts, and is targeted at finding correspondence points between the texts. In those methods we can identify two distinct stages. A generational stage that produces "candidate" correspondences and a filtering stage that sifts out bogus correspondences. The generational stage is usually based on a simple heuristic (matching n-grams, homograph tokens or cognates) and the whole method relies more on the efficacy of the filtering stage, which has also a wider ranger of different techniques: signal processing techniques (Church [6]), statistical filters (Ribeiro et al [33, 32] and algorithmic filters (Ildefonso and Lopes [19]).

In this dissertation, I depart from both trends, by introducing a methodology that, unlike lexical methods, is robust, and unlike robust methods, provides a powerful generative stage that renders the filtering stage — which has been the focus of robust methods — unnecessary. This new methodology is described in the next chapter.

3 . A new approach

This chapter presents a new method for parallel texts alignment that contrasts with previous work in several ways. The first section introduces an important shift of goals from the methods presented in chapter 2, and the second section provides an outline of the method proposed in this dissertation, which is explained in the remainder sections.

3.1 Separation of concerns

Simard et al [37], Davis et al [9], Melamed [28, 29], Ribeiro et al [32, 19] and Ribeiro [31] have used cognates as lexical cues for alignment. However, the number of cognates and loan words is highly dependent on the languages of the texts being aligned as noted by Melamed [29] and confirmed by the results of the evaluation carried by Bilbao et al [1] on the impact of cognates on alignment.

Melamed [29] suggests using a bilingual lexicon in addition to cognate matching to increase the number of correspondences. I suggest that we go one step further and use *only* a bilingual lexicon, removing cognate matching from the alignment process. Cognates should be extracted using a separate program and then inserted into the bilingual lexicon to be used for alignment. This separation makes alignment computationally lighter and allows manual verification of extracted cognates to avoid false friends.

Kay and Röscheisen [22], Chen [4], Fung and Church [11] and Fung and McKeown [12] infer (by different methods) a bilingual lexicon as part of the alignment process and use that lexicon to establish correspondences between words in the two parallel texts. Following the same reasoning as above, alignment should not be entangled with lexicon inference and instead it should rely solely on an *externally managed* bilingual lexicon. We can use the method of Ribeiro et al [36, 35] and Ribeiro [31]) to extract multi-word translation equivalents and use them for alignment. With external extraction we have full control of the lexicon that is used for alignment, meaning that we can perform manual verification of the extracted translation equivalents. Furthermore, lexicons inferred as a by-product of alignment only contain pairs of words (for performance reasons) in contrast with the multi-word translation equivalents that are extracted by the standalone programs.

To summarize, the separation of alignment from cognate/lexicon extraction presents three advantages: (1) we get richer bilingual lexicons because we can use more sophisticated methods for extraction, (2) we have control over the lexicon, meaning that we can remove any bogus entries, and (3) the alignment is computationally lighter.

In previous work it is assumed that combining multiple techniques (cognates, homographs, internal and external lexicons) to find correspondences is better than just using one of them. As explained above, using only an externally managed dictionary, we can get the same or better results. This separation of concerns marks a change of goals from previous work.

3.2 Method outline

The method produces both an alignment and a correspondence map. The *golden diagonal* mentioned earlier in section 2.2.2 is used as a crude alignment to obtain the initial correspondence map. Then, the method iterates over two steps: compute an optimal alignment from the correspondence map and, find new correspondences using the alignment as guide – the new correspondences are added to the correspondence map.

Because the map grows at each iteration, the alignment computed from the correspondence at each iteration is at least as good as the one from the previous iteration. The loop terminates when the alignment stops improving.

Correspondences are obtained using a method dubbed *neighborhood method* — section 3.4.2 — from a list of occurrence vectors that are obtained beforehand. These vectors contain the location in the parallel texts of each occurrence of known translation equivalents in both texts. Those translation equivalents are obtained externally and supplied as pairs of single- or multi-word expressions of any size.

An alignment is obtained from a correspondence map by selecting a set of occurrences that complies with the monotonicity constraint and that provides maximal coverage. The coverage of an alignment is, roughly speaking, the amount of text within the segments that are aligned.

At the outmost level the method works as follows:

1. Obtain occurrence vectors of translation equivalents in the parallel texts.
2. Generate a map of correspondences between pairs of occurrences obtained in step 1 using the *golden diagonal* as a rough guide.
3. Select the alignment with maximal coverage from the occurrence map.
4. Obtain new correspondences using the alignment obtained in step 3 as guide and add those correspondences to the map.
5. Repeat steps 3 and 4 until the alignment stops improving, i.e. until the coverage stops increasing.

The next sections describe each of these steps in detail.

offset	suffix
0	ABRACADABRA
1	BRACADABRA
2	RACADABRA
3	ACADABRA
4	CADABRA
5	ADABRA
6	DABRA
7	ABRA
8	BRA
9	RA
10	A

(a) Unsorted suffix array

LCP	offset	suffix
0	10	A
1	7	ABRA
4	0	ABRACADABRA
1	3	ACADABRA
1	5	ADABRA
0	8	BRA
3	1	BRACADABRA
0	4	CADABRA
0	6	DABRA
0	9	RA
2	2	RACADABRA

(b) Sorted suffix array and LCP array

Figure 3.1: Unsorted (a) and sorted (b) suffix arrays of text "ABRACADABRA". The offset column in the figures contains the positions in the text where each suffix starts. Note that the suffixes presented are not stored in memory individually; they can be obtained from the text and the offset column in the figure. The offset column is what we refer to as the *suffix array*.
The LCP column indicates the length of the longest common prefix between each suffix and the previous — the longest common prefixes are underlined in the suffixes.

3.3 Obtaining occurrence vectors

This section describes how to efficiently locate translation equivalents in parallel texts by taking advantage of suffix arrays with LCP information.

3.3.1 Suffix and LCP Arrays

A *suffix array* (Manber and Myers [26]) for a given text is an array that contains all the suffixes in that text. Hereafter, when we use the term *suffix array* we refer to a *sorted suffix array*. Figure 3.1 presents the suffix array for the text "ABRACADABRA" — we use this unusual text because it is short and contains a lot of repetitions. The LCP column in figure 3.1 indicates the length of the longest common prefix between each suffix and the previous one.

Manber and Myers [26] describe an algorithm to sort suffix arrays in $O(N)$ expected time (being N the length of the text).

3.3.2 Segments and occurrences

A text *segment* is represented as a pair (l,u), corresponding to the lower bound and upper bounds of the segment as shown in figure 3.2; l is the offset of the first character within the segment and u is the offset of the first character after the segment.

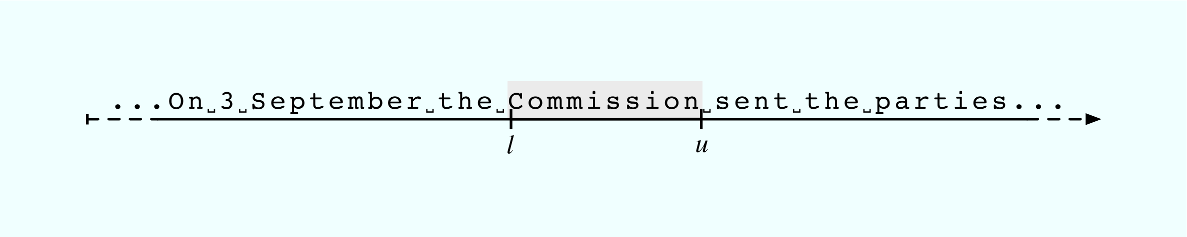


Figure 3.2: Example of a segment.

We define two auxiliary functions

$$\mathrm{lb}((l,u)) = l \quad \text{and} \quad \mathrm{ub}((l,u)) = u$$

These functions are used to obtain the lower and upper bounds when the segment is represented as a variable. For example, if we have $s = (1,2)$, then $\mathrm{lb}(s)$ is 1 and $\mathrm{ub}(s)$ is 2.

The function len gives the number of characters within the segment:

$$\mathrm{len}((l,u)) = u - l$$

A segment can contain any text, thus we use the term *occurrence* to emphasize that the text within the segment is some specific expression. For example, the segment in figure 3.2 is an occurrence of the word "Commission".

3.3.3 Locating translation equivalents in parallel texts

In this subsection we restate the initial problem of locating translation equivalents in parallel texts as the problem of locating terms in a text. The latter problem is addressed in the next subsection.

Translation equivalents are *pairs of terms*, represented as (t_x, t_y), that are translation of each other, being t_x a term of the same language as text X and t_y a term of the same language as text Y. Table 3.1 presents some English-Portuguese translation equivalents that are part of the lexicon used in the experiments with the prototype implementation of this method. Each term in the table is identified by a unique integer value, t_x for the English terms and t_y for the Portuguese terms.

For each pair of terms (t_x, t_y) in the lexicon we want to obtain a pair of occurrence vectors $(\mathbf{o_x}, \mathbf{o_y})$ that will be used later to obtain correspondences as described in section 3.4.2. The vector $\mathbf{o_x}$ should contain all the occurrences of term t_x in the text X, sorted by their position in the text. Likewise, vector $\mathbf{o_y}$ should contain all the occurrences of term t_y in text Y, sorted by their position. The problem is symmetrical and we use the same procedure for obtaining each of these vectors — hereafter we will refer to this procedure as *lookup* and it will be explained in subsection 3.3.4.

As we can observe in table 3.1, each English term t_x may be paired with a number of Portuguese terms t_y and vice versa. A more compact representation, without repeating the terms, is obtained by splitting the table into three tables. A table $\mathbf{t_x}$ of English terms

$$\mathbf{t_x} = (t_{x:1}, t_{x:2}, \ldots)$$

a table $\mathbf{t_y}$ of Portuguese terms

$$\mathbf{t_y} = (t_{y:1}, t_{y:2}, \ldots)$$

and a table $\mathbf{p}$ of pairings (t_x, t_y). Note that the latter only contains pairs of term identifiers, not the terms themselves.

See table 3.2 for an example containing the English terms that appear on table 3.1. Note that the terms in the tables are sorted. Also note that we have added LCP information to this table. The table of terms is compressed by removing the longest common prefix from each term as shown in table 3.3. This compression has two significant consequences on performance: it reduces the memory usage (32% reduction of the current lexicon) and because more terms fit in a single memory page it reduces memory page faults, improving the speed of execution. Also note, that this compression comes at absolutely no added cost because the implementation uses the LCP information of the terms to avoid re-comparing common prefixes, thus they would never be used even if they were included in the table — this compression is an consequence of the design of the algorithm rather than the way around. The details of how the LCP information is used to avoid re-comparing common prefixes are very technical and they are not included in the description of the algorithm that follows. However, compression was mentioned because it is an important advantage towards the use of large lexicons.

The *lookup* function takes two parameters: a table of terms[1] and a suffix array of the text. It returns *a vector of occurrences for each term in the table* that occur at least once in the text.

We perform $lookup(\mathbf{t_x}, X)$ to find terms of table $\mathbf{t_x}$ in text X and $lookup(\mathbf{t_y}, Y)$ to find terms of table $\mathbf{t_y}$ in text Y. The two lookups are completely independent of each other and may be executed concurrently if multiple processors are available.

3.3.4 Looking up terms in a text

In the previous subsection we restated the initial problem of locating translation equivalents as a twofold symmetrical problem of looking up a list of terms $\mathbf{t}$ in a text T.

Consider that we have the suffix array $\mathbf{a}$ for T with LCP information (we may create it with the method described by Manber and Myers [26]). Also, consider that the list of terms $\mathbf{t}$ is sorted.

[1] the terms are accessed one by one in a sequential manner, therefore we can use a list in the implementation

t_x	English term	t_y	Portuguese term
	...		...
3218	japanese government	77814	governo de o japão
65934	japaneses	147411	japonesas
65934	japaneses	147410	japoneses
65934	japaneses	147409	nipónicas
65934	japaneses	147408	nipónicos
8515	jenson tungsten ltd	83244	a empresa jenson tungsten ltd
65933	jerusalem	147407	jerusalém
65932	jesuit	147406	jesuíta
65931	jesuits	147405	jesuítas
65930	jesus	147404	jesus
70196	jesus christ	157670	jesus cristo
65929	jew	147403	judeu
65928	jews	147402	judeus
8671	johnson mathey	83406	a johnson mathey
15158	joint acp - ec ministerial trade committee	90598	comité ministerial misto acp - ce para as questões comerciais
38893	joint commission	115401	comissão mista
37852	joint committee	115401	comissão mista
37852	joint committee	120999	comité misto
43184	joint committee composed	120998	comité misto composto
37617	joint committee of the eea	93130	comité misto de o eee
37497	joint committee on road transport	115021	comité paritário de os transportes rodoviários
37447	joint committee provided for	114969	comité misto previsto
37442	joint committee referred to	114965	comité misto referido
37439	joint committee referred to in article	114960	comité misto referido em o artigo
15160	joint consultative committee	90601	comité consultivo misto
29660	joint council	106495	conselho conjunto
2529	joint declaration	77109	declaração comum
50501	joint declaration annexed	104339	declaração comum anexa
9684	joint declaration annexed hereto	84466	declaração comum em anexo
27622	joint declaration annexed to decision	104336	declaração comum anexa a a decisão
	...		...

Table 3.1: Some English-Portuguese translation equivalents. The table is sorted by the English term and the terms have been lowercased. At the time of writing, the lexicon has 98740 pairs of terms that have been manually verified.

t_x	LCP	English term
		...
3218	9	japanese government
65934	8	japaneses
8515	1	jenson tungsten ltd
65933	2	jerusalem
65932	2	jesuit
65931	6	jesuits
65930	4	jesus
70196	5	jesus christ
65929	2	jew
65928	3	jews
8671	1	johnson mathey
15158	2	joint acp - ec ministerial trade committee
38893	6	joint commission
37852	11	joint committee
43184	15	joint committee composed
37617	16	joint committee of the eea
37497	17	joint committee on road transport
37447	16	joint committee provided for
37442	16	joint committee referred to
37439	27	joint committee referred to in article
15160	8	joint consultative committee
29660	8	joint council
2529	6	joint declaration
50501	17	joint declaration annexed
9684	25	joint declaration annexed hereto
27622	26	joint declaration annexed to decision
		...

Table 3.2: Table of English terms. This table only contains terms that are present in table 3.1. The LCP column shows the length of the longest common prefix between each term and the previous. The LCPs are underlined.

t_x	LCP	English term suffix
		...
3218	9	government
65934	8	s
8515	1	enson tungsten ltd
65933	2	rusalem
65932	2	suit
65931	6	s
65930	4	s
70196	5	_christ
65929	2	w
65928	3	s
8671	1	ohnson mathey
15158	2	int acp - ec ministerial trade committee
38893	6	commission
37852	11	ttee
43184	15	_composed
37617	16	of the eea
37497	17	n road transport
37447	16	provided for
37442	16	referred to
37439	27	_in article
15160	8	nsultative committee
29660	8	uncil
2529	6	declaration
50501	17	_annexed
9684	25	_hereto
27622	26	to decision
		...

Table 3.3: Compressed version of table 3.2 obtained by removing the common prefix between each term and the previous. The space character that occurs at the beginning of some suffixes was replaced by an underscore to make it visible.

For each term t in **t** we want to locate all suffixes s in **a** that have t as a prefix:

```
function LOOKUP($\mathbf{t},\mathbf{a}$)
    $L \leftarrow \{\}$                                          ▷ the list of occurrence lists
    $t \leftarrow$ first term in t
    $s \leftarrow$ first suffix in a
    repeat
        if $t$ is prefix of $s$ then
            $\ell \leftarrow$ length of $t$
            $\mathbf{o_t} \leftarrow$ empty list                      ▷ the list of occurrences of term $t$
            $s_t \leftarrow s$
            repeat
                $pos \leftarrow$ offset of $s_t$ in the text
                append $(pos, pos+\ell)$ to $\mathbf{o_t}$ ▷ add occurrence of $t$ at the position $pos$ to the list
                $s_t \leftarrow$ next suffix in a
            until $lcp(s_k, s_{k-1}) < \ell$ or we have run through all suffixes in a
            add $\mathbf{o_t}$ to $L$
            $t \leftarrow$ the next term in t
        else if $t$ is lexicographically lower than $s$ then
            $t \leftarrow$ the next term in t
        else                                         ▷ $t$ is lexicographically higher than $s$
            $s \leftarrow$ the next suffix in a
        end if
    until we have run through all terms of t or all suffixes of a
    return $L$
end function
```

This algorithm runs in linear time $O(TextSize + TableSize)$, making it suitable for large lexicons, and it does not impose any limitation to the length of the terms. Furthermore, because it handles the texts as a sequence of bytes we meet the objective listed in section 1.2 of being as general as possible with regard to the languages of the texts being aligned.

3.4 Obtaining a correspondence map

This section describes a strategy to find correspondences between occurrences of an expression and occurrences of its translation that follows the reasoning used to solve an exercise that I dubbed as the *Champollion exercise*, explained in the next subsection.

3.4.1 The Champollion exercise

Champollion was the French linguist that was able to translate parts of the text carved in the Rosetta Stone (figure 3.3) in the first half of the 19th century. The text in the Rosetta

Stone was repeated in three writing systems: hieroglyphics, demotic script and ancient Greek. Champollion was fluent in ancient Greek and used that text to decipher the meaning of Egyptian hieroglyphs. King Ptolemy is mentioned several times in the Greek text and Champollion noticed that a set of symbols surrounded by an oval occur in roughly the same positions in the hieroglyphic text. Champollion reasoned that the symbols inside the oval probably denote Ptolemy.

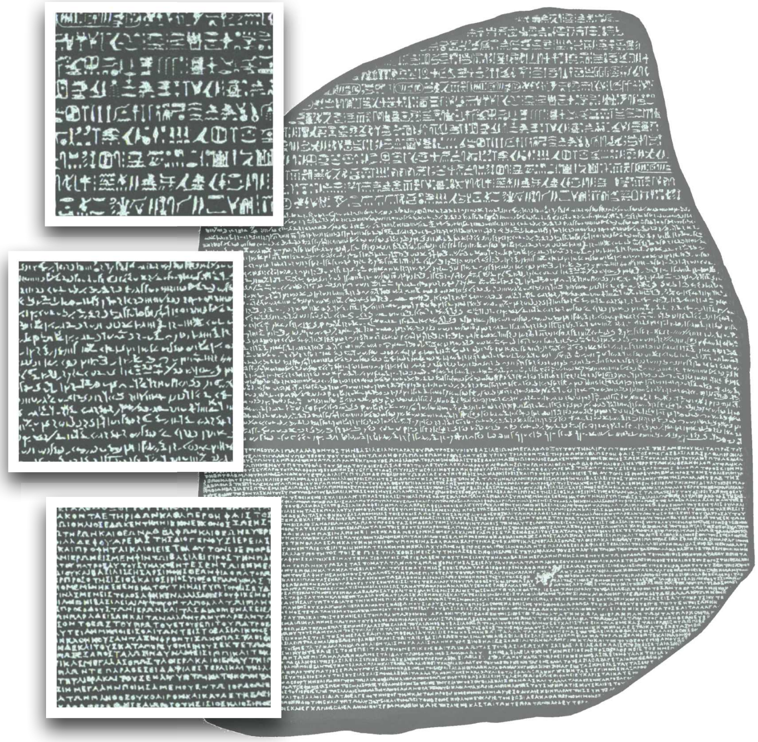


Figure 3.3: A picture of the Rosetta Stone with enlarged sections of each of the parallel texts: ancient hieroglyphics at the top, demotic script and ancient Greek at the bottom.
Adapted from the website of the European Space Agency (ESA). The original picture (without the enlarged sections) is available at: `http://esamultimedia.esa.int/images/Science/rosetta_stone_50.jpg`.

The Champollion exercise consists of the following: pick one word from a text and it's translation from a parallel text, for example "Commission" and the Portuguese word "Comissão". Pretend that all other words in the texts are undecipherable. If the chosen pair of words occurs the same number of times in both texts, then we tend to assume that the occurrences correspond pairwise. However, if the words have different frequencies, then we must look at the positions where they occur and try to figure out which occurrences should correspond to each other, based on their relative positions in the texts. For example, if both "Commission" and

"Comissão" occur in the first paragraph only once, we could assume a correspondence between these two occurrences. The method described in this section follows this reasoning, without relying on any particular concept of textual unit like *paragraph* or *sentence*. Figure 3.4 presents an instance of the problem.

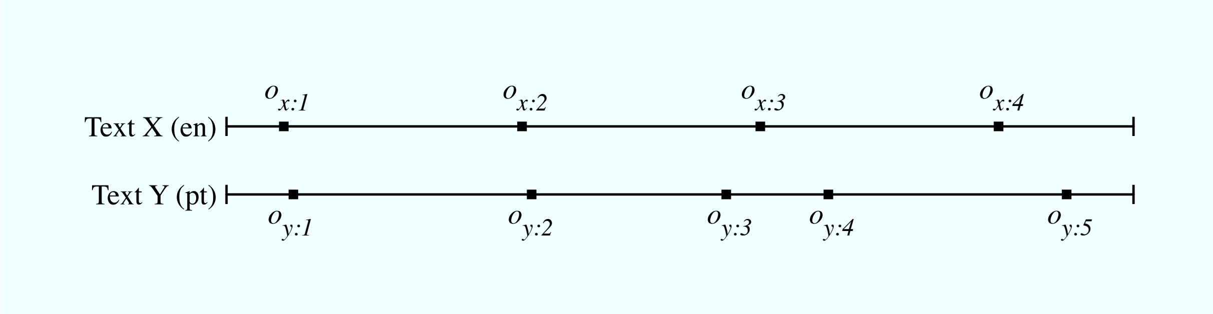


Figure 3.4: A pair of texts represented as line segments. Though the lengths of the texts are different, the lines were scaled to have the same length. The black rectangles represent occurrences of the word "Commission" in the X (English) text and it's translation, "Comissão", in the Y (Portuguese) text. In the English text the word "Commission" was replaced by the pronoun "it" in one place but not so in the Portuguese text, thus the number of occurrences of each word is different.
The question posed by the Champollion problem is: looking at this representation, which occurrences in X and Y can we assume to correspond?

My reasoning when doing the Champollion exercise was to assume a correspondence between occurrences that were roughly at the same position in both texts. But, even if the occurrences are slightly apart, we can still match them if that pair is *isolated* from the other occurrences. Figure 3.5 helps clarifying this reasoning.

Our mind is well adapted to handle problems involving patterns, so we can decide on a lot of different situations without a deep reasoning. However, computers require strict instructions so we must define the *isolation* concept with a formula. The next subsection presents one simple formulation of the isolation concept — though other formulations could be devised.

> He said that the quick decipherment enabled him 'to avoid the systematic errors which invariably arise from prolonged reflection.' You get better results, he argued, by not thinking too much. — Carl Sagan about Champollion's work

3.4.2 The neighborhood method

A simple decision rule is that we may assume correspondence between a pair of occurrences if both are within a *neighborhood* of acceptance that is determined from the distance to the

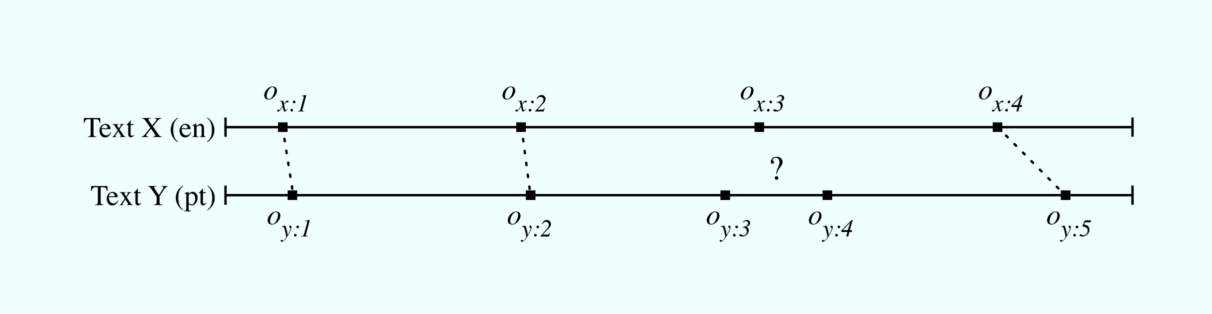


Figure 3.5: Correspondences manually established according to the isolation criteria. Occurrences $o_{x:1}$ and $o_{y:1}$ are distant from $o_{x:2}$ and $o_{y:2}$, thus we have no problem assuming they correspond. The same holds for $o_{x:2}$ and $o_{y:2}$. The occurrences $o_{x:4}$ and $o_{y:5}$ are not so close to each other, but we can, arguably, match them as well. However, I could not decide which occurrence of Y corresponds to $o_{x:3}$, even though $o_{x:3}$ is closer to $o_{y:3}$ than $o_{x:4}$ is to $o_{y:5}$. Actually, $o_{x:3}$ corresponds to $o_{y:3}$ and the occurrence $o_{y:4}$ corresponds to the pronoun "it" in the English text, but according to the rules of the game we don't known about "it".

closest occurrences. I defined that neighborhood as a bilateral restriction, shown in figure 3.6. A neighborhood interval $h_{x:i}$ is defined for each occurrence $o_{x:i}$ in X as

$$h_{x:i} = \left[\frac{u_{x:(i-1)} + l_{x:i}}{2}, \frac{u_{x:i} + l_{x:(i+1)}}{2} \right]$$

where $l_{x:i} = \text{lb}(o_{x:i})$ and $u_{x:i} = \text{ub}(o_{x:i})$. In the same way, a neighborhood interval $h_{y:j}$ is defined for each occurrence $o_{y:j}$ in Y as

$$h_{y:j} = \left[\frac{u_{y:(j-1)} + l_{y:j}}{2}, \frac{u_{y:j} + l_{y:(j+1)}}{2} \right]$$

For the first occurrence in text X (having $i = 1$) we consider $\text{ub}(o_{x:i-1}) = 0$, that is, the beginning of the text. For the last occurrence ($i = n$) we consider $\text{lb}(o_{x:n+1}) = L_x$, being L_x the length of text X. Thus, the neighborhood interval for the first occurrence in text X is

$$h_{x:1} = \left[\frac{0 + l_{x:1}}{2}, \frac{u_{x:1} + l_{x:2}}{2} \right]$$

and for the last occurrence is

$$h_{x:n} = \left[\frac{u_{x:(n-1)} + l_{x:n}}{2}, \frac{u_{x:n} + L_x}{2} \right]$$

The first and last occurrences in text Y are computed in a similar way (omitted).

For now, let's assume there is a function $\alpha : [0, L_x] \rightarrow [0, L_y]$ that maps each position in X to a position in Y and a function $\beta : [0, L_y] \rightarrow [0, L_x]$ that maps each position in Y to a position in X. These functions are defined later in subsection 3.4.3, but for now we need to know that both are monotonically increasing functions.

Because they are monotonically increasing we have that

$$x_1 < x_2 \Rightarrow \alpha(x_1) \leq \alpha(x_2) \quad \text{and} \quad y_1 < y_2 \Rightarrow \beta(y_1) \leq \beta(y_2)$$

To simplify the explanation we can abuse the notation and use $\alpha(h_{x:i})$ and $\beta(h_{y:j})$ as shorthands for

$$\alpha(h_{x:i}) = [\alpha(\text{lb}(h_{x:i})), \alpha(\text{ub}(h_{x:i}))] \quad \text{and} \quad \beta(h_{y:j}) = \left[\beta(\text{lb}(h_{y:j})), \beta(\text{ub}(h_{y:j}))\right]$$

Finally, the function that is used to compute the correspondence map is defined as

$$\text{correspond}(o_{x:i}, o_{y:j}) = \begin{cases} \text{yes, if } o_{x:i} \subseteq \beta(h_{y:j}) \text{ and } o_{y:j} \subseteq \alpha(h_{x:i}) \\ \text{no, otherwise} \end{cases}$$

and the map $\mathbf{M}$ of correspondence between the texts X and Y is

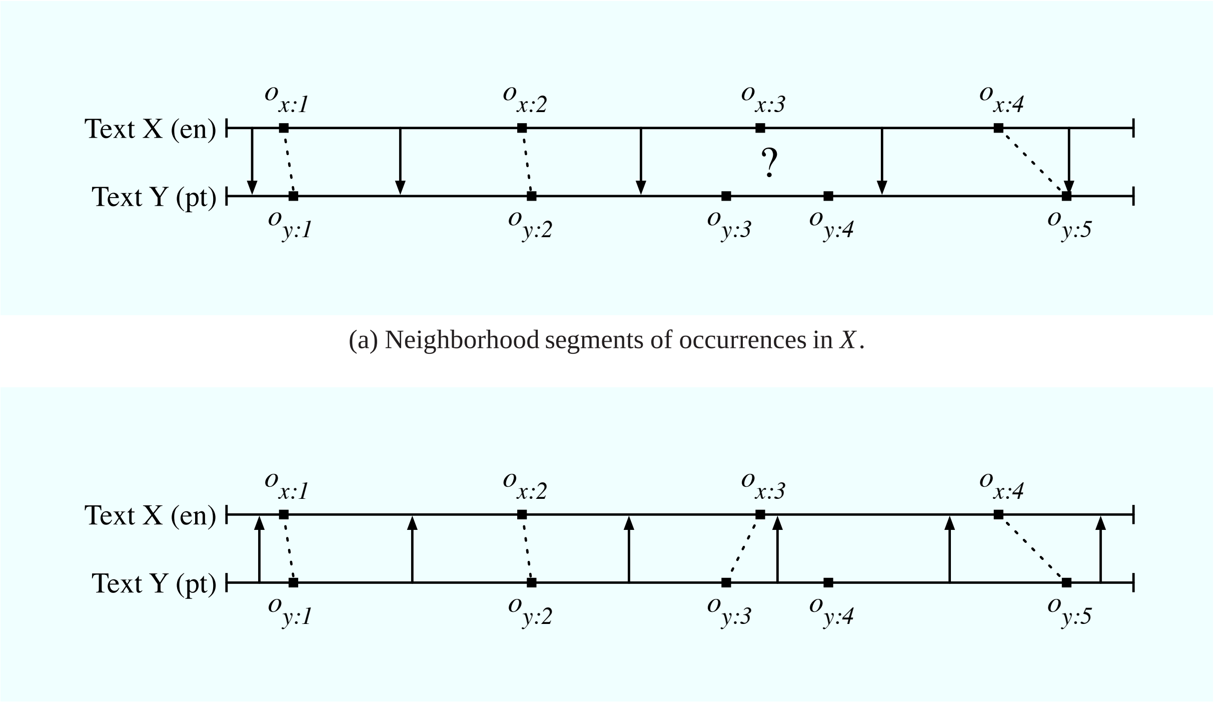
$$\mathbf{M} = \bigcup \left\{ (o_{x:i}, o_{y:j}) \in \mathbf{o_{x:p}} \times \mathbf{o_{y:p}} : \text{correspond}(o_{x:i}, o_{y:j}) \forall p \in \mathbf{p} \right\}$$

where $\mathbf{o_{x:p}}$ and $\mathbf{o_{y:p}}$ are the occurrence vectors of each pair of terms p in the set of pairings $\mathbf{p}$ that was described in section 3.3.4. The occurrence vectors are obtained by the methods explained in section 3.3.

3.4.3 Using alignment as a guide

The correspondence between a segment $s_x = (l_x, u_x)$ of text X and a segment $s_y = (l_y, u_y)$ of text Y may be represented in a cartesian plane as the rectangle in figure 3.7. The lower-left and upper-right vertices of the rectangle are the points (l_x, l_y) and (u_x, u_y) respectively.

If we plot all correspondences of a pair of texts in a cartesian plane we observe that they line up diagonally, as shown in figure 3.8. This "golden diagonal" has been "discovered" long ago and previous work described in chapter 2 explores it in one way or another. For example, the main hypothesis of the length-based methods described in 2.1.1 is that the length of parallel texts tends to be proportional, and the work by Ribeiro et al described in section 2.2.2 uses linear regression lines to filter noisy correspondences and those lines are very close to the "golden diagonal". However, those methods do not check the isolation of occurrences to assess



(a) Neighborhood segments of occurrences in X.

(b) Neighborhood segments of occurrences in Y.

Figure 3.6: Neighborhood segments used to find correspondences. Two occurrences are assumed to correspond if each is within the neighborhood interval defined by the other.
The bounds of the neighborhood intervals of each occurrence in X are represented in figure (a) as arrows pointing to their expected positions (given by α) in text Y. Figure (b) shows the bounds of neighborhood intervals of occurrences in Y as arrows pointing to their expected positions (given by β) in X. Note that if the lines representing the texts had not been scaled to have identical lengths, the arrows would be skewed.
The functions α and β are described in the next section.

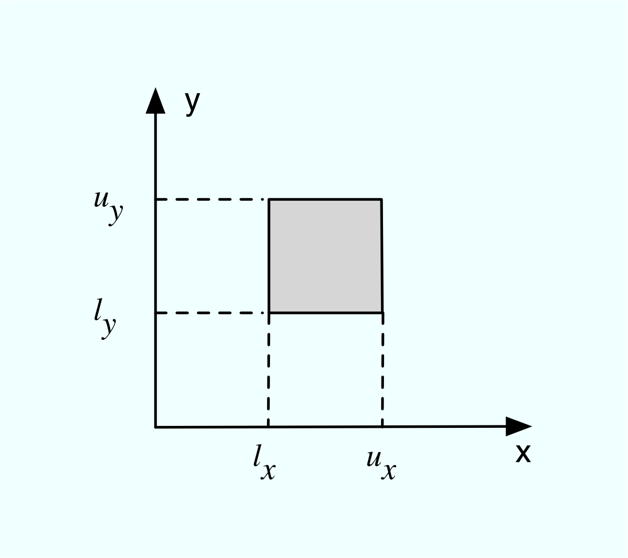


Figure 3.7: Representation of correspondence in the cartesian plane as a rectangle. Given a segment $s_x = (l_x, u_x)$ of text X and a segment $s_y = (l_y, u_y)$ of text Y the bottom left and top right vertices of the correspondence rectangle are defined by the points (l_x, l_y) and (u_x, u_y) respectively.

the confidence of the correspondences.

A *polygonal chain* is a piecewise linear curve. It is defined as a sequence of points (vertices) and the curve is obtained by connecting consecutive vertices with line segments.

An alignment of two parallel texts may be used to define a monotone[2] polygonal chain **c** as

$$\mathbf{c} = ((0,0),(l_{x:1},l_{y:1}),(u_{x:1},u_{y:1}),(l_{x:2},l_{y:2}),(u_{x:2},u_{y:2}),\cdots,(l_{x:n},l_{y:n}),(u_{x:n},u_{y:n}),(L_x,L_y))$$

where (lb_{xi}, lb_{yi}) and (ub_{xi}, ub_{yi}) are the lower and upper vertices of the correspondence rectangle as shown in figure 3.10. L_x and L_y are the lengths of X and Y respectively.

The previously mentioned functions α and β are closely related to the monotone polygonal chain **c** obtained from alignment. Function α computes the ordinate that intercepts **c** at the given abscissa and function β computes the abscissa that intercepts **c** at the given ordinate.

[2] with x and y coordinates ever increasing

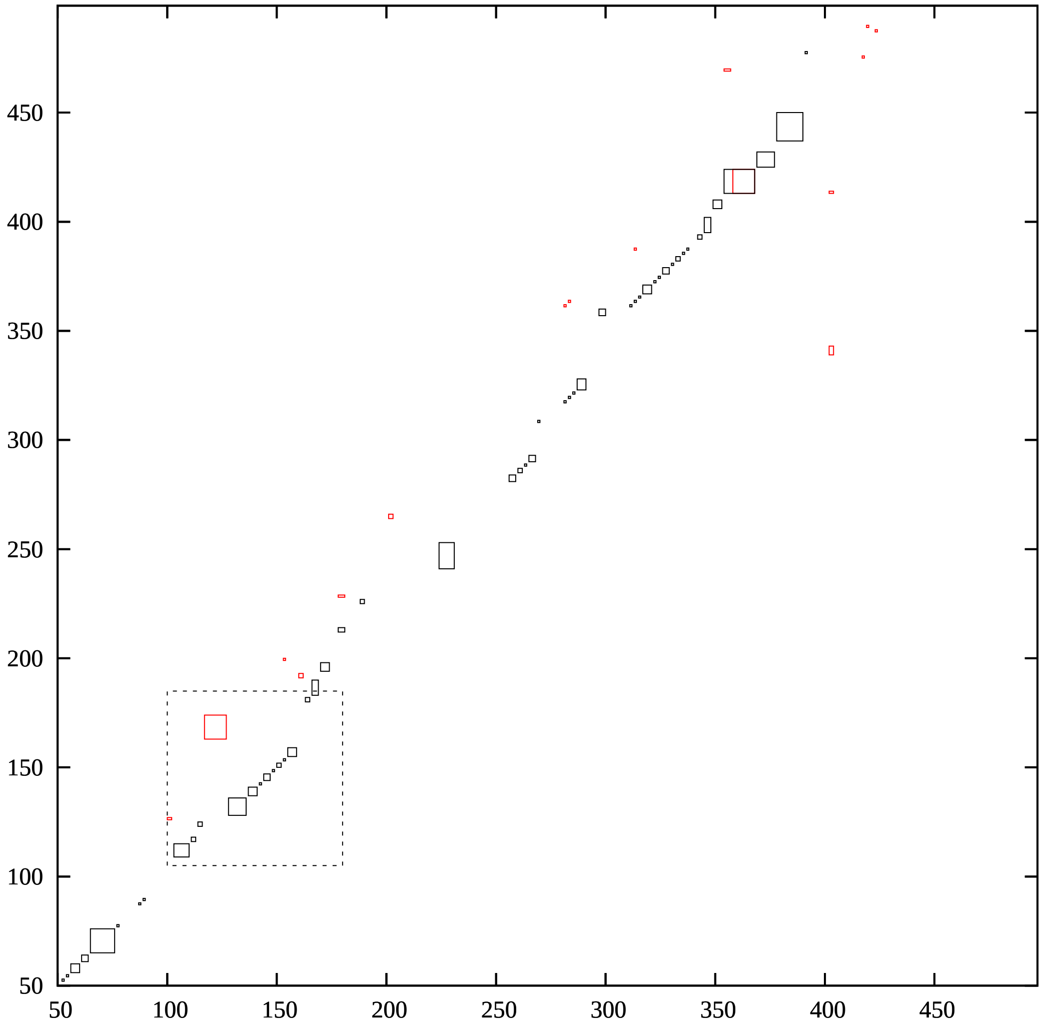


Figure 3.8: Correspondences between segments of a pair of texts. Each correspondence is represented as a rectangle (as described in figure 3.7). The red correspondences are not part of the selected alignment (selection is discussed in section 3.5). The area delimited by the dashed rectangle is shown in figure 3.9 with greater detail.

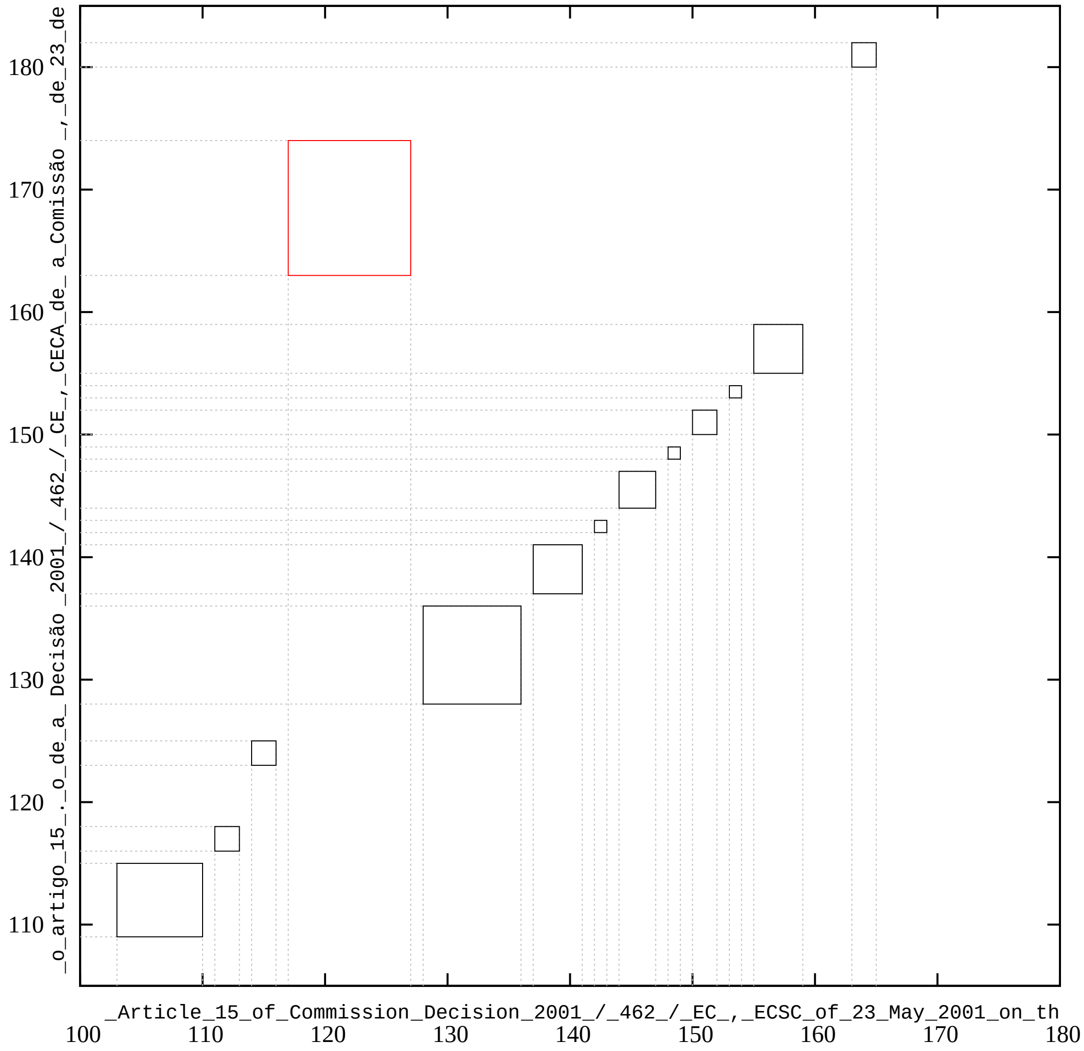


Figure 3.9: Detail of the area delimited by the dashed rectangle in figure 3.8 showing word order changes in translation. The correspondence in red is not part of the selected alignment (selection is discussed in section 3.5).

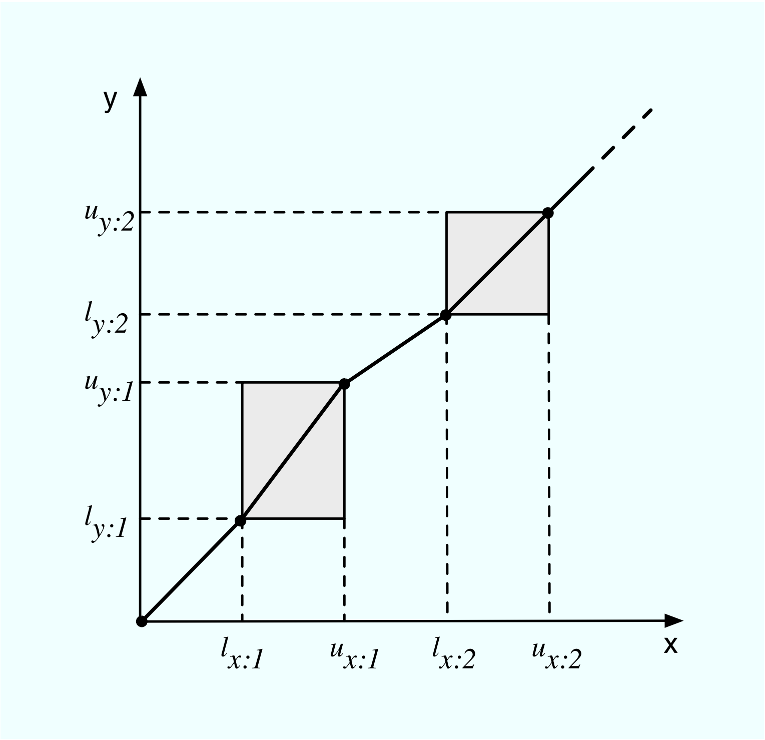


Figure 3.10: A monotone polygonal chain obtained from alignment.

Function α is defined in terms of the *monotone polygonal chain* **c** as

$$\alpha(x) = \begin{cases} x(l_{y:1}/l_{x:1}) & 0 \leq x < l_{x:1} \\ l_{y:1} + (x - l_{x:1})(u_{y:1} - l_{y:1})/(u_{x:1} - l_{x:1}) & l_{x:1} \leq x < u_{x:1} \\ u_{y:1} + (x - u_{x:1})(l_{y:2} - u_{y:1})/(l_{x:2} - u_{x:1}) & u_{x:1} \leq x < l_{x:2} \\ \dots & \\ u_{y:n} + (x - u_{x:n})(L_y - u_{y:n})/(L_x - u_{x:n}) & u_{x:n} \leq x < L_x \end{cases}$$

and function β is defined as:

$$\beta(y) = \begin{cases} y(l_{x:1}/l_{y:1}) & 0 \leq y < l_{y:1} \\ l_{x:1} + (y - l_{y:1})(u_{x:1} - l_{x:1})/(u_{y:1} - l_{y:1}) & l_{y:1} \leq y < u_{y:1} \\ u_{x:1} + (y - u_{y:1})(l_{x:2} - u_{x:1})/(l_{y:2} - u_{y:1}) & u_{y:1} \leq y < l_{y:2} \\ \dots & \\ u_{x:n} + (y - u_{y:n})(L_x - u_{x:n})/(L_y - u_{y:n}) & u_{y:n} \leq y < L_y \end{cases}$$

These functions were used in previous subsection to compute the *expected* position in the other text for some position in one text, i.e. $\alpha(x)$ gives the position in text Y that, according to the current alignment, is expected to correspond to the position x in text X.

This concludes the section explaining how to obtain a correspondence map from the occurrence vectors obtained in section 3.3, using an alignment as a guide.

3.5 Obtaining an alignment from a correspondence map

This section presents an algorithm for selecting a set of non-crossing correspondences with maximal coverage from a correspondence map. As we have seen in chapter 1, a set of non-crossing correspondences is an alignment. Next we will see what *coverage* is, and then we discuss why the coverage is a good criteria for selection. The algorithm for obtaining the selection is presented at the end of this section.

The coverage of an alignment A is the portion of the texts that is within the segments $(s_x, s_y) \in A$. More precisely, the coverage of an alignment A is given by

$$\text{coverage}(A) = \frac{\sum \text{len}(s_x) + \text{len}(s_y)}{L_x + L_y} \quad , (s_x, s_y) \in A \tag{3.1}$$

being L_x and L_y the lengths of texts X and Y.

The Longest Sorted Sequence Algorithm (LSSA) described by Ildefonso and Lopes [19] also selects an alignment from a set of correspondences, although their correspondences are pairs of text positions (points in the cartesian plane) instead of pairs of text segments (rectangles in the cartesian plane). They view the correspondences that form an alignment as a sequence of points with increasing x and y coordinates. The criteria for selecting the more reliable sequence is the length of the sequence, measured in terms of the number of points. If there are two or more sequences having maximum length, they select the one having the rightmost (last) point.

The reasoning behind the LSSA is that, the most probable alignment is the one that includes more correspondence points. The maximal coverage criteria, on the other hand, acknowledges that not all correspondences are equally reliable, and thus, they should not be all equally weighted when deciding the best alignment. From observation, the correspondences between small segments, like correspondences between occurrences of punctuation characters, are not as reliable as the occurrences between occurrences of large terms, as for example a correspondence between "rubber products manufacturing" and the Portuguese translation "a indústria transformadora de produtos de borracha". We hypothesize that correspondences between larger segments are more reliable. According to this hypothesis, the more reliable alignment is the one with maximal coverage.

We define the boolean operator precedes for two correspondences as

$$(s_{x:1}, s_{y:1})\,\mathrm{precedes}(s_{x:2}, s_{y:2}) = \begin{cases} \text{true} & \text{if} \quad \mathrm{ub}(s_{x:1}) \leq \mathrm{lb}(s_{x:2}) \wedge \mathrm{ub}(s_{y:1}) \leq \mathrm{lb}(s_{y:2}) \\ \text{false} & \text{otherwise} \end{cases}$$

The algorithm for selecting the alignment A with maximal coverage from a correspondence map M works as follows:

```
function SELECT($M$)
    $A \leftarrow \{\}$
    for all $m \in M$ do
        $M_p \leftarrow \{m_p \in M : m_p \,\mathrm{precedes}\, m\}$
        $A_p \leftarrow$ SELECT($M_p$)
        if coverage$(m \cup A_p) >$ coverage$(A)$ then
            $A \leftarrow m \cup A_j$
        end if
    end for
    return $A$
end function
```

3.6 Iterative improvement

Because the alignment is computed from an occurrence map and the occurrence map is computed using an alignment as a guide, we must provide an initial alignment to bootstrap the loop. We use the *golden diagonal* as the initial rough alignment.

Alternatively we can supply an existing alignment, and the method will try to improve it.

The neighborhood method behaves in a way that allows us to start with a bad alignment and still, obtain the correct correspondences. This behavior is consequence of the method being extremely cautious about the correspondences it makes. It is designed to rather not make a correspondence if there is any evidence that the correspondence might be wrong. When the guiding alignment is bad, only the large neighborhood segments will intercept each other having the respective occurrences within the interception. The size of the neighborhood is determined by the proximity between occurrences of the same term, therefore, less frequent terms are more likely to have larger neighborhoods. If we give a bad alignment as a guide, the method will make few correspondences, but they are likely to be correct. Then, these correct correspondences are used to obtain an alignment and in the next iteration the method will make a lot more correspondences. In the experiments with the prototype, two behaviors were observed regarding the number of correspondences obtained at each iteration: if the initial alignment is bad — for example if we give a diagonal that starts at $(0, L_y/3)$ and ends at (L_x, L_y), consisting of an alignment that discards a third of text Y — the number of occurrences in the first iteration is less than in the second iteration; if we give a reasonable alignment — like the *golden diagonal* — then the number of iterations in the first iteration is greater than the second. This shows that the global iterative algorithm finds it's way even if we point it in the wrong direction.

The loop stops when the alignment stops improving, i.e. when the coverage of current alignment is not larger than the previous. Because coverage takes values in the interval $[0, 1]$ the loop is guaranteed to stop if the coverage ever reaches 1 (total correspondence). In practice the alignment stabilizes after few iterations (4 iterations on average in the experiments with the JRC-Acquis corpus).

3.7 Evaluation of alignment results

There are two mainstream methodologies for evaluating the quality of an alignment: (a) comparing the alignment against a "golden standard" previously created by hand (Melamed [30], Véronis and Langlais [40] and Chiao et al [5]), and (b) manual verification of the correctness of an alignment (Bilbao et al [1]). Method (a) is suitable for evaluating several alignments of the same corpus when they are produced by different programs. However, the creation of a "golden standard" alignment is a very demanding/expensive task and, as a consequence is not adequate for having it repeated for every new parallel corpus addressing other subject matters. Therefore, method (b) is preferable when only one alignment/program is being evaluated – which is the case.

According to the method described by Bilbao et al [1] the evaluator looks at a number of aligned segments and grades the correctness of each pair according to the ratio of the number of words that are correctly aligned and the number of words in the longest of the two segments. Thus, a segment that is graded with 1 is a correct translation, and a segment with grade 0 is totally incorrect. The precision of the whole alignment is computed as the average of segment grades.

The alignment method described in the previous sections was used for aligning the European Constitution corpus (part of the OPUS project [39]) using a lexicon with 62k English-Portuguese pairs of terms that have been automatically extracted and manually verified. The alignment was evaluated by my supervisor, Professor Gabriel Pereira Lopes, according to the method presented in the previous paragraph.

A total of 2424 segments were evaluated with an average precision of 76.56%. There are notably few errors resulting from incorrect correspondences obtained by the neighborhood method described in subsection 3.4.2. Most alignment errors could be avoided if the lexicon contained more single-word translations — most of the terms in the lexicon are phrases.

Bilbao et al [1] report a maximum[3] alignment precision of 75.46% for the alignment method described by Ildefonso and Lopes [19]. Although the evaluation method is the same, the results of the two evaluations cannot be meaningfully compared because there are too many variables that may have an impact on the result — evaluations were conducted on different corpora, the alignment granularity was different, etc. Nevertheless, the precision of the alignments in this experiment allows us to say that this method, even with a small lexicon, can produce alignments with a quality that is roughly comparable to the quality of those obtained by the method of Ildefonso and Lopes [19]. Moreover, the precision of term translations automatically extracted from corpora aligned with the method by Ildefonso and Lopes was typically 20% (unpublished work, personal communication). In a recent experiment using the European Constitution corpus aligned by the new alignment method, the precision of the extracted term translations is about 70%. Because the extraction method used in both experiments is nearly identical, we conclude that the new alignment method should be credited for some part of this improvement.

[3] the alignment method takes a treshold parameter that must be adjusted for each pair of languages; depending on the threshold parameter the alignment precision varies between 53.37% and 75.46%

4. Conclusion

The proposed approach meets the objectives listed in section 1.2, taking advantage of bilingual dictionaries to find correspondences in parallel texts and being as general as possible with regard to the languages of the texts by not relying on any linguistic concept — the texts are seen as a stream of Unicode characters.

The lexicon used in the evaluation experiment described in section 3.7 contained about 62k English-Portuguese pairs of terms, which is a relatively small lexicon for an approach that relies exclusively on a lexicon to obtain correspondences. A larger lexicon is expected to improve the precision of alignment because most low-graded segments in that experiment contain words that were not part of the lexicon. Nevertheless, the precision of the alignment is already similar to the precision of the method of Ildefonso and Lopes [19] which relies on cognates and homographs.

The evaluation result shows that it is possible to obtain an alignment at least as good as the one obtained by the method of Ildefonso and Lopes, while relying exclusively on an external lexicon, thus supporting the thesis that lexicon inference should not be embedded in the alignment process. Instead, the alignment should be part of a larger iterative scheme, in line with extraction of translation equivalents and human validation of the extracted equivalents, as depicted in figure 4.1.

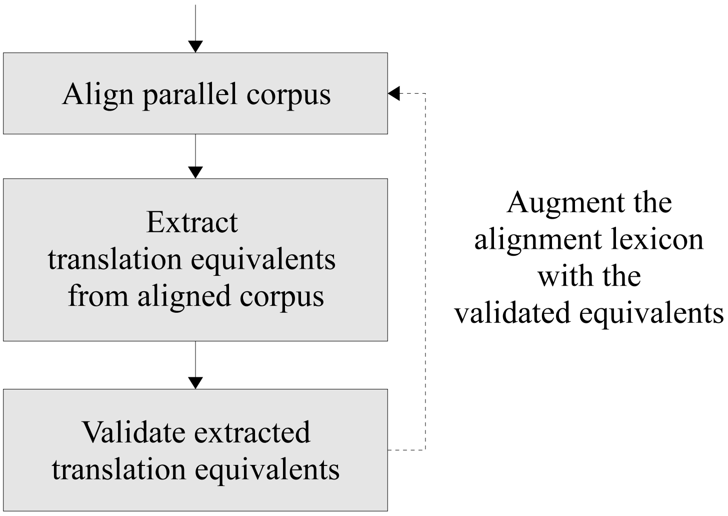

Align parallel corpus
Extract translation equivalents from aligned corpus
Validate extracted translation equivalents
Augment the alignment lexicon with the validated equivalents

Figure 4.1: The "Big Picture": the alignment process as part of a larger iterative scheme that improves alignment quality at each iteration.

4.1 Summary of contributions

This work adds several important contributions to the state of the art:

1. An algorithm that uses suffix arrays with LCP information to locate multi-word translation equivalents in a text.

2. A new method for obtaining correspondences that effectively avoids most of the noise from the search space.

3. An algorithm to obtain an alignment with the maximal coverage from a set of correspondences.

4.1.1 Human-machine synergy

I'm convinced that, with human help, machines can do a lot better than by themselves let alone. The new approach described is prepared to use a large bilingual lexicon of single- and multi-word translation equivalents. This enables a human-machine synergy towards improving the alignment quality over time by augmenting the bilingual lexicon.

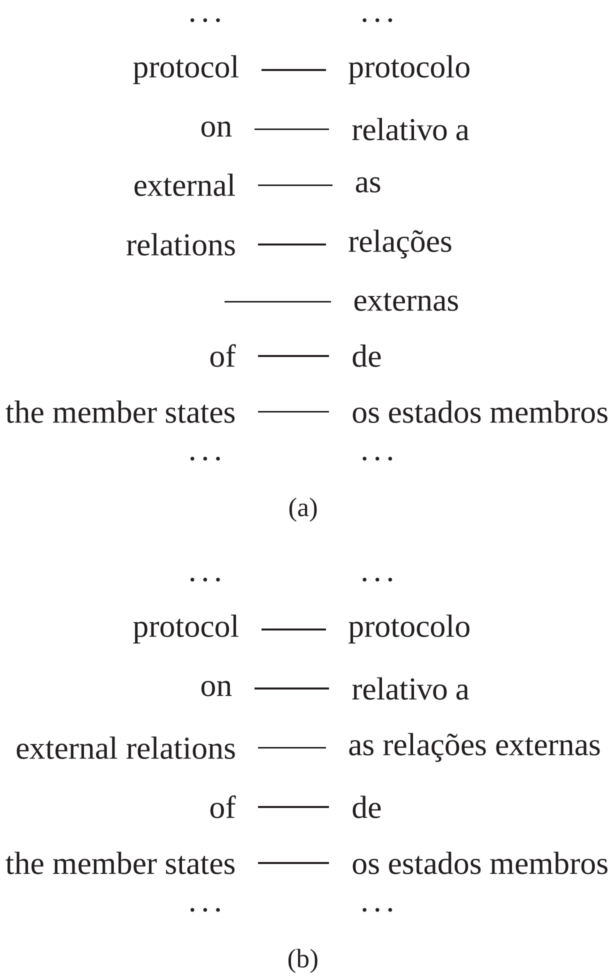
We can (ab)use the bilingual lexicon to prevent frequent alignment mistakes, in particular those resulting from different word order within cohesive expressions, by introducing single- and multi-word translations into the dictionaries. Figure 4.2 gives an example.

While not being a full-fledged solution to the word order problem the bilingual lexicon provides a quick fix for frequent expressions.

Moreover, an aligned parallel corpus can be an input for an autonomous extractor of translation equivalents. Experiments made (not published yet) on top of English and Portuguese texts from the European Constitution, aligned with the prototype implementation of the method here described, enabled the extraction of approximately 500 translations of single and multi-word terms occurring more than 5 times with a 71% precision. The number of extracted single and multi-word term translations occurring more than twice is approximately 8000, and these were not yet evaluated. Reuse of validated translation entries will necessarily improve alignment quality and enable the identification of translation errors."

4.1.2 Data driven

The method relies more on data and less on algorithms. In particular, it does not perform cognate matching nor lexicon inference, relying solely on external bilingual lexicons for obtaining correspondences. This approach marks an important departure from most previous methods, which perform automatic extraction of some kind: in those methods, the alignment errors caused by wrongly extracted word pairs are mechanically repeated over all aligned texts and there is nothing one can do about it because the extraction is done within the alignment itself.



(a)

(b)

Figure 4.2: Figure (a) shows a frequent misalignment. Introducing the pair "external relations"–"as relações externas" into the dictionary, prevents that misalignment as show in figure (b). Despite not being a full-fledged solution for the problem of word order changes, we can fix frequent expressions in this way.

For example, the methods that perform cognate matching like Simard et al [37], Ribeiro et al [32], Ribeiro [31], Church [6] and Melamed [29, 28] are all vulnerable to *false friends*. Those methods will align false friends that pass their cognaticity test if they occur in similar positions in the texts. In our case we only align words or expressions that are in the dictionary.

Takes advantage of sophisticated methods for translation equivalent extraction If there is no dictionary for a given pair of languages, we can align using only verbatim passages and then perform an automatic extraction of translation equivalents using the method suggested by Ribeiro et al [31]. We can repeat this alignment–extraction cycle several times to augment the dictionary until the alignment produced has an acceptable granularity — the alignment gets more fine-grained as the number of correspondences between words or expressions increases.

4.1.3 Obtaining more and more reliable correspondences

The method for obtaining correspondences described here outperforms the methods described in previous work. The number of obtained correspondences depends mostly on the size of the translation equivalents table. In the experiments conduced with a dictionary of about 60 thousand entries there was less than 10% of the words left without correspondence on average. On the other hand, the observed number of bogus correspondences is very low.

When compared with the signal-processing techniques used by Church [6] to obtain a clean signal from dotplots, or the complex SIMR pattern-recognition algorithm described by Melamed [28], or the K-vec method by Fung and Church [11] and the successor DK-vec by Fung and McKeown [12], the proposed method is simpler and computationally lighter (linear runtime and space). The method suggested by Ribeiro et al of using terms with equal frequencies on both texts as alignment candidates is definitely simple and computationally light, however it does not provide as many candidates.

4.1.4 No need to filter

Compared to methods that are genetically closer to the approach here described — including Church [6], Ribeiro et al [33, 34, 32], Ribeiro [31] and Ildefonso and Lopes [19] — we may highlight one important departure: I have shifted the emphasis from the filtering stage to the generative stage. This turned out to be a good decision because the improved candidate generation makes the filtering stage unnecessary, contributing to the overall computational lightness of the method.

4.1.5 Iterative improvement of alignment

The methods described in Kay and Röscheisen [22], Ribeiro et al [33, 34, 32], Ribeiro [31] and Ildefonso and Lopes [19] at some point assert that some set of selected correspondences are correct in order to proceed finding more correspondences. However, *none of those methods* *re-evaluates past assertions in face of new information*.

In other words, the new correspondences always "agree" with older ones. As a consequence, erroneous correspondences prevent good correspondences from being generated in their neighborhood.

The re-evaluation performed by the selection algorithm at each iteration decides towards the best global alignment based on the hypothesis that any incorrect correspondences will "disagree" with most of their neighbors and therefore, the best alignment can be selected by choosing the one with maximal coverage.

4.2 Future work

Given the computational lightness of the prototype and the encouraging results, I consider that the new methods presented here deserve further research, in particular to see how they perform when applied to pairs of unrelated languages or to other kinds of text, like the Bible, medical documentation, technical manuals, etc. Alignment between pairs of unrelated languages poses additional challenges. In particular non Indo-European languages like Arabic, Hindi, Chinese and Japanese.

It is pertinent to investigate the asymptotic limitations of this lexicon-based approach when the lexicon becomes richer, or in other words, assess the quality of alignment as lexicon becomes larger. Eventually, adding more entries to a lexicon will not produce noticeable improvements in the alignment.

Functional words account for most of the noisy correspondences. One possibility to avoid noisy correspondences is to have a two phase alignment. The first phase should not use functional words. The alignment from the first phase would be refined in the second phase by fitting functional words within the correspondences from phase one.

Bibliography

[1] Víctor Bilbao, Gabriel Pereira Lopes, and Tiago Ildefonso. Measuring the impact of cognates in parallel text alignment. In Amilcar Cardoso Carlos Bento and Gael Dias, editors, *2005: Portuguese Conference on Artificial Intelligence, Proceedings*, pages 338–343. IEEE Computer Society, 12 2005.

[2] Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, Robert L. Mercer, and Paul S. Roossin. A statistical approach to language translation. In *COLING*, pages 71–76, 1988.

[3] Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. Aligning sentences in parallel corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 169–176, Morristown, NJ, USA, 1991. Association for Computational Linguistics.

[4] F. Chen, Stanley. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA, 1993. Association for Computational Linguistics.

[5] Yun-Chuang Chiao, Olivier Kraif, Dominique Laurent, Thi Minh Huyen Nguyen, Nasredine Semmar, François Stuck, Jean Véronis, and Wajdi Zaghouani. Evaluation of multilingual text alignment systems: the ARCADE II project. In *5th international Conference on Language Resources and Evaluation - LREC'06*, Genoa/Italy, 05 2006.

[6] Ward Church, Kenneth. Char_align: a program for aligning parallel texts at the character level. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 1–8, Morristown, NJ, USA, 1993. Association for Computational Linguistics.

[7] Joaquim Ferreira da Silva, Gael Dias, Sylvie Guilloré, and José Gabriel Pereira Lopes. Using localmaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. In P. Barahona, editor, *Progress in Artificial Intelligence: 9th Portuguese Conference on AI, EPIA'99, Évora Portugal September 1999, Proceedings. Lecture Notes in Artificial Intelligence*, volume 1695, pages 113–132. Springer-Verlag, 1999.

[8] Ido Dagan and Kenneth W Church. Termight: Identifying and translating technical terminology. In *In Proceedings of the 4th Conference on Applied Natural Language Processing*, pages 34–40, 1994.

[9] Mark W. Davis, Ted E. Dunning, and William C. Ogden. Text alignment in the real world: improving alignments of noisy translations using common lexical features, string matching strategies and n-gram comparisons. In *Proceedings of the seventh conference on*

European chapter of the Association for Computational Linguistics, pages 67–74, Dublin, Ireland, 1995. Morgan Kaufmann Publishers Inc.

[10] Gaël Dias. *Extraction Automatique d'Associations Lexicales à partir de Corpora*. PhD thesis, Universidade Nova de Lisboa and Université d'Orléans, December 2002.

[11] Pascale Fung and Ward Church, Kenneth. K-vec: a new approach for aligning parallel texts. In *Proceedings of the 15th conference on Computational linguistics*, pages 1096–1102, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[12] Pascale Fung and Kathleen Mckeown. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. In *In Proceedings of the First Conference of the Association for Machine Translation in the Americas, 81–88*, pages 81–88, 1994.

[13] A. Gale, William and W. Church, Kenneth. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 177–184, Morristown, NJ, USA, 1991. Association for Computational Linguistics.

[14] W. A. Gale, K. W. Church, and D. Yarowsky. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439, 1993.

[15] William A. Gale and Kenneth W. Church. Identifying word correspondences in parallel texts. In *In Proceedings of the Fourth DARPA Speech and Natural Language Workshop*, pages 152–157, 1991.

[16] William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.

[17] Masahiko Haruno and Takefumi Yamazaki. High-performance bilingual text alignment using statistical and dictionary information. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 131–138, Santa Cruz, California, 1996. Association for Computational Linguistics.

[18] John Hutchins. Machine translation: History. In Keith Brown, editor, *Encyclopedia of Languages and Linguistics*, volume 7, pages 375–383. Elsevier, Oxford, 2nd edition, 2006.

[19] Tiago Ildefonso and Gabriel Lopes. Longest sorted sequence algorithm for parallel text alignment. *Computer Aided Systems Theory –EUROCAST 2005*, pages 81–90, 2005.

[20] Pierre Isabelle. Bi-textual aids for translators. In *University of Waterloo*, pages 76–89, 1992.

[21] Martin Kay and Martin Röscheisen. Text-translation alignment. Technical Report P90-00143, Xerox Palo Alto Research Center, Palo Alto, CA, 1988.

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

[23] Judith Klavans and Evelyne Tzoukermann. The BICORD system: combining lexical information from bilingual corpora and machine readable dictionaries. In *Proceedings of the 13th conference on Computational linguistics*, pages 174–179, Morristown, NJ, USA, 1990. Association for Computational Linguistics.

[24] Philipp Koehn. Europarl: a parallel corpus for statistical machine translation. In *MT summit X, the tenth machine translation summit*, pages 79–86. Phuket, Thailand, 2005.

[25] Julian Kupiec. An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 17–22, Morristown, NJ, USA, 1993. Association for Computational Linguistics.

[26] Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[27] Dan Melamed, I. A word-to-word model of translational equivalence. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 490–497, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

[28] Dan Melamed, I. Bitext maps and alignment via pattern recognition. *Comput. Linguist.*, 25(1):107–130, 1999.

[29] I. Dan Melamed. A portable algorithm for mapping bitext correspondence. In *ACL-35: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 305–312, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

[30] I. Dan Melamed. Manual annotation of translational equivalence: The blinker project. Technical Report IRCS-98-07, Institute for Research in Cognitive Science, 1998.

[31] António Ribeiro. *Parallel Texts Alignment for Extraction of Translation Equivalents*. PhD thesis, Universidade Nova de Lisboa, Lisboa, 2002.

[32] António Ribeiro, Gael Dias, G. P. Lopes, and João Tiago Mexia. Cognates alignment. In Bente Maegaard, editor, *Proceedings of the Machine Translation Summit VIII (MT Summit VIII), Santiago de Compostela, Spain, September 18-22, 2001*, pages 287–292. European Association of Machine Translation, 09 2001.

[33] António Ribeiro, G. P. Lopes, and João Tiago Mexia. Linear regression based alignment of parallel texts using homograph words. In Werner Horn, editor, *ECAI 2000: Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, 2000 August 20-25*, pages 446–450. IOS PRESS, 08 2000.

[34] António Ribeiro, Gabriel Lopes, and João Mexia. Using confidence bands for parallel texts alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 432–439, Hong Kong, 2000. Association for Computational Linguistics.

[35] António Ribeiro, Gabriel Pereira Lopes, and João Mexia. Extracting translation equivalents from portuguese-chinese parallel texts. *Studies in Lexicography (Korea, Republic of)*, 11(1):118–194, 2001.

[36] António Ribeiro, José Gabriel Pereira Lopes, and João Mexia. Extracting equivalents from aligned parallel texts: Comparison of measures of similarity. In *Proceedings of the International Joint Conference IBERAMIA/SBIA*, pages 339–349, São Paulo, Brazil, 2000.

[37] M Simard, G Foster, and P Isabelle. Using cognates to align sentences in parallel corpora. In *In Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 67–81, 1992.

[38] Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Daniel Varga. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. *CoRR*, abs/cs/0609058, 2006.

[39] J. Tiedemann and L. Nygaard. The OPUS corpus - parallel and free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004.

[40] Jean Véronis and Philippe Langlais. Evaluation of parallel text alignment systems. In Jean Véronis, editor, *Parallel Text Processing: Alignment and Use of Translation Corpora*, chapter 19. Kluwer, Dordrecht, 2000.

[41] Thomas Wannacott and Ronald Wannacott. *Introductory Statistics*. John Willey & Sons, New York Chichester Brisbane Toronto Singapore, 5th edition edition, 1990.

[42] Dekai Wu. Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 80–87, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[43] Dekai Wu. Grammarless extraction of phrasal translation examples from parallel texts. In *In Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 354–372, 1995.

[44] Dekai Wu. Alignment. In Robert Dale, Hermann Moisl, and Harold Somers, editors, *Handbook of Natural Language Processing*, pages 415–458. Marcel Dekker, New York, July 2000.

[45] Dekai Wu and Xuanyin Xia. Learning an english-chinese lexicon from a parallel corpus. In *In Proceedings of the First Conference of the Association for Machine Translation in the Americas*, pages 206–213, 1994.