Aligning Clauses in Parallel Texts

Sotiris Boutsis and **Stelios Piperidis**
Institute for Language and Speech Processing - ILSP
Artemidos & Epidavrou 151-25, Athens, Greece
tel:+301 6800959, fax:+301 6854270
email: {sboutsis, spip}@ilsp.gr

National Technical University of Athens - NTUA

Abstract

This paper describes a method for the automatic alignment of parallel texts at clause level. The method features statistical techniques coupled with shallow linguistic processing. It presupposes a parallel bilingual corpus and identifies alignments between the clauses of the source and target language sides of the corpus. Parallel texts are first statistically aligned at sentence level and then tagged with their part-of-speech categories. Regular grammars functioning on tags, recognize clauses on both sides of the parallel text. A probabilistic model is applied next, operating on the basis of word occurrence and co-occurrence probabilities and character lengths. Depending on sentence size, possible alignments are fed into a dynamic programming framework or a simulated annealing system in order to find or approximate the best alignment. The method has been tested on a small English-Greek corpus consisting of texts relevant to software systems and has produced promising results in terms of correctly identified clause alignments.

Introduction

The availability of large collections of texts in electronic form, has given rise to a wide range of applications aiming at the elicitation of linguistic resources such as translation dictionaries, transfer grammars and retrieval of translation examples (Dagan et al., 1991; Matsumoto et al., 1993), or even the building of fully-blown machine translation systems (Brown et al., 1990). The purpose of this paper is to describe a technique for extracting translation correspondences at bellow sentence level by employing statistical techniques coupled with shallow linguistic processing catering for the segmentation of sentences into clauses.

Statistical processing has proved powerful for the extraction of translation equivalences at sentence and intra-sentence level. Brown et al. (1991) described a method based on the number of words contained in sentences. The general idea is that the closer in length two sentences are, the most likely they are to align. Moreover, certain anchor points and paragraph markers are considered. Dynamic programming and HMMs are pipelined to produce alignments at sentence level. The method has been applied to the Hansard-Corpus, achieving an accuracy of 96%-97%. Gale and Church (1991) proposed a method that relies on a simple statistical model of character lengths. The model is based on the observation that longer sentences in one language tend to be translated into longer sentences in the other language while shorter ones tend to be translated into shorter ones. A probabilistic score is assigned to each pair of proposed sentence pairs, and a dynamic programming framework calculates the most probable alignment. Although the apparent efficacy of the Gale-Church algorithm is undeniable and validated on different pairs of languages, it faces problems when handling complex alignments(1-0, 1-2, 2-2).

Simard et al. (1992) argue that a small amount of linguistic information is necessary in order to overcome the inherited weaknesses of the purely statistical techniques. They proposed using cognates, which are pairs of tokens of different languages sharing "obvious" phonological or orthographic and semantic properties, since these are likely to be used as mutual translations. Papageorgiou et al. (1994) proposed a generic alignment scheme invoking surface linguistic information coupled with information about possible unit delimiters depending on the level at which alignment is sought. Each unit, sentence, clause or phrase, is represented by the sum of its content part of speech (POS) tags. The results are then fed into a dynamic programming framework that computes the optimum alignment of text units.

Brown (1988) uses a probabilistic measure to estimate word similarity of two languages in the context of statistically-based machine translation. Kay and Roescheisen (1993) present an algorithm for aligning bilingual texts on the basis of internal evidence. Processing is performed in many iterations and each new iteration uses the results of the previous one in order to calculate more accurate word and sentence correspondences. In

each iteration, processing consists of calculating correspondences between sentences on the basis of their relative positions, and then calculating word correspondences on the basis of word co-occurrences in related sentences. The Dice coefficient is used as the similarity measure between words of two languages in an attempt to secure the correctness of the alignment of parallel texts at sentence level. Kitamura and Matsumoto (1995) have used the same Dice coefficient to calculate the word similarity between Japanese-English parallel corpora. Single word correspondences have also been investigated by Gale and Church (1991b) using a statistical evaluation of contingency tables. Piperidis et al. (1997) and Boutsis and Piperidis (1996) describe methods for extracting single and multi-word equivalences based on a parallel corpus statistically aligned at sentence level and employing a similarity metric along the lines of the Dice coefficient with comparable performance.

Collocational correspondences have been studied by Smadja (1992) and Smadja et al. (1996), in an attempt to find translation patterns for continuous and discontinuous collocations in English and French. Meaningful collocations are first extracted in the source language while their corresponding French ones are found by calculating the mutual information between instances of the English collocation and various single word candidates in English-French aligned corpora. Recent work has broadened the scope identifying correspondences between word sequences. Kupiec (1993) proposes a method for extracting translation patterns of noun phrases from English-French parallel corpora. The corpus is tagged at part-of-speech (POS) level and then finite-state recognizers specified by regular expressions defined in terms of POS categories detect noun phrases on either side. Probabilities of correspondences are then calculated using an iterative EM-like algorithm. Kumano and Hirakawa (1994) presuppose an ordinary bilingual dictionary and non-parallel corpora, attempting to find bilingual correspondences in a Japanese-English setting at word, noun phrase and unknown word level. Extending previous work, Kitamura and Matsumoto (1996) apply the Dice coefficient on word sequence correspondence extraction.

This paper describes a method for the automatic alignment of parallel texts at clause level. Texts are first aligned at sentence level using statistical techniques. Part-of-speech tagging takes place next annotating each word form with the appropriate part of speech. Processing in this step and the next one is monolingual, so each language side of the text is treated independently of the other. Surface syntactic analysis is performed next on the basis of regular grammars. Shallow parsing results in the recognition of clauses. Statistical processing follows taking into account different sources of information, aiming at identifying intra-sentence alignments formed by the clauses of the parallel sentences of the bitext. The method caters for alignments of type 1-0, 1-1, 1-2, 2-1, and 2-2. A first pass through the text computes occurrence and co-occurrence probabilities for content words on both language sides. A probabilistic score, expressing the probability that a clause (or a pair of clauses) of the source language is translated into a clause (or a pair of clauses) of the target language, is computed on the basis of the previously calculated word probabilities, and a model of character lengths. Possible clause alignments are examined by a dynamic programming framework deciding on the best alignment. Avoiding combinatorial explosion requires that large sentences be channeled into a module that approximates the optimal alignment through simulated annealing, operating in polynomial time. EM iterative training caters for the estimation of the model's parameters, given the lack of hand-aligned training material. The overview of the processing is pictured in Figure 1.

Test Corpus

The corpus used to develop and test the proposed algorithms consists of text from the HP-VUE software platform documentation set. The Greek text contains 35726 wordforms and the English text 28872. The number of different words is 4512 for the Greek text and 3219 for the English text. The richer morphology of the Greek language accounts for the approximately 30% difference between these two figures.

Text Handling

Recognizing and labeling surface phenomena in the text is a necessary prerequisite for most Natural Language Processing (NLP) systems. In order to be able to make full use of the corpus, texts should be rendered in an appropriate form. To this end, parallel texts are normalized and handled. In the framework of the presented method, basic text handling is performed with the use of a Multext-like tokeniser, (Di Christo et al., 1995). Identification of word boundaries, sentence boundaries, abbreviations etc. takes place. Following common practice, the tokeniser makes use of a regular-expression based definition of words, coupled with downstream precompiled lists for the Greek and English language and simple heuristics. This proves to be quite successful in recognizing sentences and words effectively.

Sentence Alignment

Alignment consists in establishing correspondence links between units in a bilingual text. At this stage, the method aligns input text at sentence level. Processing caters for sentence substitution (one sentence translates into one), deletion (a sentence is not translated at all), insertion (a sentence with no equivalent in the source text

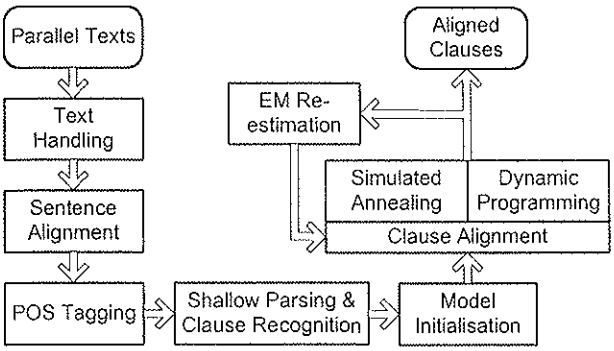

Figure 1: Processing Overview

is introduced by the translator), contraction (two consecutive sentences translate into one), expansion (one sentence translates into two) and merging (two sentences translate jointly into two).

The heart of the alignment scheme, employed at this stage, is a method for aligning sentences based on a simple statistical model of character lengths, (Gale and Church, 1991). The method relies on the assumption that longer sentences in the source language tend to be translated into longer sentences in the target and vice-versa. A probabilistic score is assigned to each pair of proposed sentence pairs, based on the ratio of lengths of the sentences and the variance of this ratio. This probabilistic score is used in a dynamic programming framework in order to find the maximum likelihood alignment of sentences. Additionally, following (Brown et al., 1991) certain points of the texts can be anchored thus dividing them into smaller sections that need to be aligned. Besides anchors, paragraph markers are also considered. Anchor points are specific to the text to be aligned and they usually appear in both texts. They are divided into major and minor anchors and alignment proceeds in two steps, first aligning major anchor points and then minor anchor points, followed by sentence alignment. The alignment algorithm has been tested in the setting of a multilingual text processing system and has been reported to yield accuracy between 96% and 100%, (Piperidis, 1995).

Part of speech tagging

Both English and Greek texts are analyzed morphosyntactically. The words in the parallel sentences are tagged with their corresponding POS categories. The corpus is thus represented as a bitext of tagged mutual sentence translations where every word is accompanied by its corresponding POS tag.

For Greek

Tagging with part-of-speech information for Greek takes place in two steps. First, each word is endowed with all possible tags through lexicon lookup, and then a disambiguation module decides on the most probable annotation.

Lexicon lookup operates on a morphological lexicon of modern Greek. It endows the words of the text with the characteristics found in the lexicon. The tagset used has been devised for the morphological annotation of Greek corpora and conforms to the guidelines set up by EAGLES and PAROLE, trying, at the same time, to capture the morphological peculiarities of the Greek language.

Text produced at the output of lexicon lookup is annotated with below POS information i.e. subcategorisation information for each POS category. Each wordform recognised as noun, for example, is annotated for case, number, gender etc. Ambiguous wordforms are endowed with all possible annotations. However, not all available morphological information is necessary for later processing. In addition, wordforms grammatically fully characterized with below POS information are highly ambiguous. Retaining all such information would impose a heavy burden on the disambiguation process. Experimentation has proved that performance of next stages is not seriously affected by reducing the tagset. To this end, a simplified tagset has been used helping reduce ambiguous wordforms notably. In addition, words not found in the lexicon are assigned possible tags on the basis of a probabilistic model operating on word suffixes. In case of multiple tagging, a disambiguator based on trigrams and contextual rules trained on Greek texts, suggests the tag that is most likely to be the correct, (Papageorgiou, 1996). This stage produces around 95% correct results.

For English

Tagging for English is based on mainstream statistical processing. A tagger implementing hidden markov model techniques is employed. The tagger has been trained on a large preannotated text collection and is then used to tag the HP-VUE test corpus. For training purposes, a set of technical texts annotated at POS level, drawn from the British National Corpus (BNC), has been used, (Burnard, 1995). Texts classified under the field codes: "Written: Domain: Informative: Natural and pure sciences" and "Written: Domain: Informative: Applied Science" have been selected. The size of the text collection is ca. 5,000,000 words. Text is annotated with POS tags according to the BNC tagset (Leech, 1995). This text collection is used to train the Acquilex HMM tagger (Elworthy, 1997) and estimate model parameters. After training, the HP-VUE corpus is tagged by application of the Viterbi algorithm.

Clause recognition

This stage, like the previous one, processes each language side of the text independently of the other. It aims at breaking sentences of both languages into clauses with well-defined boundaries.

In order to recognise clauses, this stage takes advantage of a shallow parser equipped with grammars for Greek and English. Syntactic analysis consists of parsing via finite state automata. Under this approach, a text can be analysed syntactically on the basis of grammars containing non-recursive rules written in the form of regular expressions. Rules are numbered in order to be applied in a certain order. The grammar is translated into finite-state automata with standard techniques (Aho et al., 1986) and automata are connected in a pipeline in order to form a cascade, which is used to annotate text in an incremental way. Each rule (regular expression) describes a specific phenomenon and higher-order rules can be expressed on the basis of the already described ones. Rules are designed to be reliable when they are applied using longest match, in order to avoid the need for disambiguation between different length instances of the same constituent type.

A basic characteristic of this method is that parsing is deterministic and no backtracking takes place. No ambiguity is produced since each automaton takes a definite decision about a constituent's existence or non-existence. This doesn't mean that ambiguities are resolved but that they are enclosed inside syntactic chunks, whose boundaries have been recognised, although their internal structure may have not been decided. Enclosure of ambiguity helps generate only one partial parse for each sentence, since ambiguity is kept local and does not cause the production of multiple parses for the whole sentence.

It should be noted that the method does not depend on the exact method adopted for clause recognision. Another system performing clause recognition could be used instead. This has also to do with the availability of the relevant linguistic processing modules. On the other hand, being aware of the complete partial parse can be very useful, if one is up to extend the method to cover other types of sub-sentence alignments (e.g. alignment of np's). It is also significant that the additional processing of shallow parsing does not impose serious speed overheads since the speed of analysis is measured in tens of hundreds of words/second. Clause boundaries for each analysed sentence are channelled into the next stages of processing. No distinction is made between different clause types. A sample output of this stage is shown in Figure 2.

[cl SEVERAL UTILITIES HELP YOU cl] [cl DIAGNOSE CONFIGURATION AND DATABASE ERRORS cl]

[cl ΠΟΛΛΑ ΒΟΗΘΗΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ ΒΟΗΘΟΥΝ cl] [cl ΝΑ ΔΙΑΓΝΩΣΕΤΕ ΣΦΑΛΜΑΤΑ ΔΙΑΜΟΡΦΩΣΗΣ ΚΑΙ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ cl]

[cl IF YOUR SYSTEM IS PROPERLY CONFIGURED cl] [cl TO AUTOMATICALLY RUN HP VUE cl] , [cl YOU WILL SEE THE HP VUE LOGIN SCREEN cl] [cl WHEN YOUR SYSTEM IS BOOTED cl]

[cl ΑΝ ΤΟ ΣΥΣΤΗΜΑ ΣΑΣ ΕΙΝΑΙ ΣΩΣΤΑ ΔΙΑΜΟΡΦΩΜΕΝΟ cl] [cl ΓΙΑ ΝΑ ΕΚΤΕΛΕΙ ΑΥΤΟΜΑΤΑ ΤΟ HP VUE cl] [cl ΘΑ ΔΕΙΤΕ ΤΗΝ ΟΘΟΝΗ ΣΥΝΔΕΣΗΣ ΤΟΥ HP VUE cl] [cl ΟΤΑΝ ΤΟ ΣΥΣΤΗΜΑ ΣΑΣ ΕΚΚΙΝΕΙ cl]

[cl IF YOU HAVE NO CONSOLE cl] , [cl YOU MUST LOG IN FROM A REMOTE SYSTEM cl]

[cl ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ cl] [cl ΠΡΕΠΕΙ ΝΑ ΕΙΣΕΛΘΕΤΕ ΑΠΟ ΕΝΑ ΑΠΟΜΑΚΡΥΣΜΕΝΟ ΣΥΣΤΗΜΑ cl]

Figure 2: Parallel text with marked clause boundaries

Translation model

Part a

In this section we present the basic translation model, which is used for the purposes of clause alignment. Let's consider two corresponding sentences of the parallel text which are translations of each other, the source sentence $\underline{S_i} = \underline{sc_{i1}}\ \underline{sc_{i2}} \ldots \underline{sc_{il}}$ and its translation into the target language $\underline{T_i} = \underline{tc_{i1}}\ \underline{tc_{i2}} \ldots \underline{tc_{im}}$ where $\underline{sc_i}$ and $\underline{tc_i}$ are clauses identified during the previous stage. We approximate sentence translation with the assumption that clauses can be translated from the source into the target language in the following ways:

A. 1-0 and 0-1, when a clause of the source or the target sentence has no equivalent clause in the other language.
B. 1-1, when a clause of the source sentence is translated into one clause of the target sentence.
C. 1-2 and 2-1, when a clause of the source is translated into two clauses of the target or two clauses of the source translate into one of the target.
D. 2-2, when two clauses jointly translate into two clauses of the other language.

We view each group of aligned sentences of the parallel text as a sequence of clause-beads (after sentence-beads in (Brown et al., 1991)) where a bead accounts for a group of clauses that align with each other according to one of the above mentioned ways. A clause-alignment

$A_i = \{ a_{i1}\ a_{i2} \ldots a_{in} \}$ for a given pair i of sentences is a set of clause-beads a_{ij} covering all clauses of the source and target sentence under the condition that each clause participates to one and only one clause-bead. Figure 3 shows a schematic example of a clause-alignment between two sentences containing four and three clauses each. Making the assumption that translation of clauses in a bead is independent of clauses belonging to other beads we seek the alignment that maximises the joint distribution:

$$\Pr(\underline{S_i}, \underline{T_i}, A_i) = \Pr(n) \prod_{j=1}^{n} \Pr(a_{ij}) \qquad (1)$$

and assuming that $\Pr(n)$ (where n is the number of beads in the alignment) is independent of S_i, T_i and n we get:

$$\Pr(\underline{S_i}, \underline{T_i}, A_i) = \varepsilon \prod_{j=1}^{n} \Pr(a_{ij}) \qquad (2)$$

ε is ignored for the rest of the analysis, since it is a multiplicative constant factor having the same value for all clause-alignments.

Part b

Finding the correct alignment requires that we estimate clause-bead probabilities $\Pr(a_{ij})$ which express the probability for the source sentence clauses of the bead to be translated into the corresponding target sentence clauses. We consider a 1-1 bead covering the source and target clauses:

$\underline{sc_{is}} = sw_{is1}\ sw_{is2} \ldots sw_{isp}$ and

$\underline{tc_{it}} = tw_{it1}\ tw_{it2} \ldots tw_{itq}$

(where sw_{isp} is the pth word of the sth clause of the ith source sentence of the parallel text etc.) A first writing of $\Pr(a_{ij})$ can be as follows:

$$\Pr(a_{ij}) = P_{1-1}\ \Pr(\underline{sc_{is}}, \underline{tc_{it}}) \qquad (3)$$

where P_{1-1} is the probability of a '1-1' clause alignment. Referring to the second factor of (3), in order to approximate $\Pr(\underline{sc_{is}}, \underline{tc_{it}})$ we take into account two parameters: a) the length of the source and target clauses and b) the source language and target language words contained in $\underline{sc_{is}}$ and $\underline{tc_{it}}$. We model the probability that source text with character length $l(\underline{sc_{is}})$ is translated into target text with length $l(\underline{tc_{it}})$ with a distribution $\Pr\left(l(\underline{sc_{is}}), l(\underline{tc_{it}})\right)$. Under the assumption that the model used by the sentence aligner ("Sentence Alignment" section , (Gale and Church, 1991)) expressing sentence alignment probabilities on the basis of character lengths is valid when applied to clause-lengths, we estimate $\Pr\left(l(\underline{sc_{is}})\, l(\underline{tc_{it}})\right)$ with the same model.

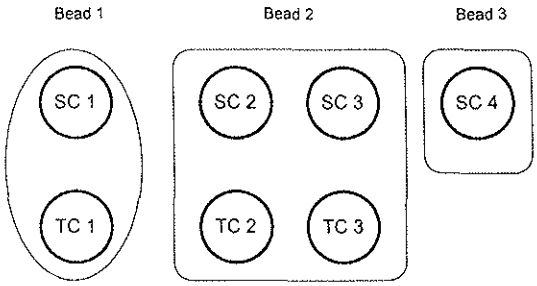

Figure 3: An alignment with three beads
(SC:Source sentence Clause
TC:Target sentence Clause)

Furthermore, we approximate clauses by unordered sets focusing on content carrying words i.e. content words, which are taken to be verbs, nouns, adjectives and adverbs. Thus, we assume that content words contribute the most to the examined probability. $\underline{tc_{it}}$ and $\underline{sc_{is}}$ are represented by the unordered sets of the content words they contain. Following that, equation (3) can be written as:

$$\Pr(a_{ij}) = P_{1-1} \cdot \Pr\left(l(\underline{sc_{is}}), l(\underline{tc_{it}})\right) \cdot \Pr\left(\{scw_{is1}, \ldots, scw_{isv}\}, \{tcw_{it1}, \ldots, tcw_{itw}\}\right) \qquad (4)$$

where scw stands for source clause content word and tcw stands for target clause content word. To approximate the third factor of Eq. (4) we assume that the content words of the source clause are independent events and the same is valid for the words of the target clause. That is:

$$\Pr(\{scw_{is1}, scw_{is2}, \ldots, scw_{isv}\}) = \Pr(scw_{is1})\ \Pr(scw_{is2}) \ldots \Pr(scw_{isv}) \qquad (5)$$

$$\Pr(\{tcw_{it1}, tcw_{it2}, \ldots, tcw_{itw}\}) = \Pr(tcw_{it1})\ \Pr(tcw_{it2}) \ldots \Pr(tcw_{itw}) \qquad (6)$$

Under this model each word of the target clause depends on zero or one word of the source clause. To il-

lustrate, let's consider the source clause $\underline{sc} = \{ scw_1, scw_2, scw_3 \}$ the target clause $\underline{tc} = \{ tcw_1, tcw_2, tcw_3 \}$ and a word alignment W_j so that tcw_1 depends on scw_1, tcw_2 depends on scw_2 while tcw_3 and scw_3 are independent events. In this case,

$$\Pr_{W_j}(\{ scw_1, scw_2, scw_3 \},\{ tcw_1, tcw_2, tcw_3 \}) =$$

$$\Pr(tcw_1, scw_1)\Pr(tcw_2, scw_2)\Pr(tcw_3)\Pr(scw_3) \quad (7)$$

given the computation of Figure 4.

Consequently, when estimating bead probability $\Pr(a_{ij})$, we need to sum probabilities over all possible word alignments W_j. This would require however to inspect an exponentially large set of possible word-alignments. Thus, we would like to approximate the sum with its biggest term. This is not feasible, either. So, a greedy-like technique is followed, which does not guarantee to find the best word alignment but usually comes up with a big enough value to distinguish between good and not so good clause alignments. The largest word-pair probabilities are selected first while probabilities of any unmatched words are taken into account next. In order to select a pair of words for Eq. (7) two heuristic conditions must be met: 1) the occurrence frequencies of the two words should not differ more than 50%, 2) their co-occurrence frequency in the bitext should not differ more than 50% from their occurrence frequencies in the texts.

In case of a non '1-1' alignment between clauses, the same model is used, where P_{1-1} is substituted by P_{1-2}, P_{2-1}, P_{2-2}, P_{1-0} and P_{0-1}. We take $P_{1-2} = P_{2-1}$ and $P_{1-0} = P_{0-1}$. The distribution on character lengths is also taken to be independent of the alignment type.

Model Training

In order to calculate clause-alignment probabilities, given the model presented in the previous section, estimations for several model parameters should be available. At this stage, parameters are estimated on the basis of simple corpus statistics. The probability of a single word of the source or target text is taken to be:

$$\Pr(w) = \frac{f(w)}{\sum_{w'} f(w')} \quad (8)$$

where the denominator of Eq. (8) is the sum of the frequencies of all words i.e. the length of the source or the target text in words. Correspondingly, the probability relating a word of the source text with a word of the target text is estimated by:

$$\Pr(sw, tw) = \frac{f(sw, tw)}{\sum_{(sw', tw')} f(sw', tw')} \quad (9)$$

For the presented application of the method, these probabilities are computed over the whole corpus. In very large texts it is adequate to estimate the probabilities in a representative large portion of the text. It would be also possible to use pre-computed probabilities from another text of the same domain, given that both texts share

$$\Pr_{W_j}(\{ scw_1, scw_2, scw_3 \},\{ tcw_1, tcw_2, tcw_3 \}) =$$

$$\Pr_{W_j}(\{ tcw_1, tcw_2, tcw_3 \} | \{ scw_1, scw_2, scw_3 \}) \Pr(\{ scw_1, scw_2, scw_3 \}) = \quad \textit{(Eq.(5), (6))}$$

$$\Pr_{W_j}(tcw_1 | \{ scw_1, scw_2, scw_3 \}) \Pr_{W_j}(tcw_2 | \{ scw_1, scw_2, scw_3 \}) \Pr_{W_j}(tcw_3 | \{ scw_1, scw_2, scw_3 \}) \cdot$$

$$\Pr(scw_1)\Pr(scw_2)\Pr(scw_3) =$$

$$\Pr(tcw_1 | scw_1)\Pr(tcw_2 | scw_2)\Pr(tcw_3)\Pr(scw_1)\Pr(scw_2)\Pr(scw_3) =$$

$$\frac{\Pr(tcw_1, scw_1)}{\Pr(scw_1)} \frac{\Pr(tcw_2, scw_2)}{\Pr(scw_2)} \Pr(tcw_3)\Pr(scw_1)\Pr(scw_2)\Pr(scw_3) =$$

$$\Pr(tcw_1, scw_1)\Pr(tcw_2, scw_2)\Pr(tcw_3)\Pr(scw_3)$$

Figure 4: Computation of $\Pr_{W_j}(\{ scw_1, scw_2, scw_3 \},\{ tcw_1, tcw_2, tcw_3 \})$

the same characteristics with respect to language use, coverage and translation.

Estimating $P_{1\text{-}1}, P_{1\text{-}2}, P_{2\text{-}2}$ and $P_{0\text{-}1}$ is less straightforward. Given the lack of training material, that is marked-up text aligned at clause level, no safe set of values can be computed for these parameters. To work around this, we first make an educated guess and then apply the EM (Expectation-Maximization) algorithm. The EM algorithm consists of two major steps: an expectation step followed by a maximization step. The expectation uses the current estimates of the parameters to process input data and the maximization provides next a new estimate of these parameters. These two steps iterate until convergence. EM is not guaranteed to converge to a global maximum; if many points of local convergence exist, the point where the method will convergence will depend on the initial parameter estimations. The initial parameter values we used and the estimated ones after the process converged are displayed in the Table 1.

If an alignment type does not occur in the output ('1-0' alignment in this case), the relevant probability takes a very small value (1E-4).

Best Clause-Alignment Selection

This stage aims at finding the best alignment between the clauses of two parallel sentences (or in the case of a non '1-1' sentence alignment e.g. '1-2', an alignment is sought between the clauses of the source sentence and the clauses of the two target sentences). Two schemes are considered, dynamic programming and simulated annealing.

Dynamic programming is a generalization of the greedy technique. It can be used to solve problems, whose solutions can be considered as a sequence of decisions. Usually dynamic programming is used to address an optimization problem, seeking the sequence of decisions giving the optimal solution. In many problems, decisions taken on the basis of local data always lead to optimal solutions; this is the case of problems solved by greedy techniques. On the other hand, there are problems, including alignment, for which this doesn't hold true. In this case one would have to generate all possible decision sequences and evaluate them. Dynamic programming can be used to exclude sub-optimal decision sequences so that they may not be considered. The principle of optimality governing dynamic programming is: "Any sub-sequence of the optimal decision sequence is optimal for the sub-problem corresponding to this sub-sequence of decisions".

Although dynamic programming is successfully applied to sentence alignment, it comes close to its limits when dealing with sub-sentence alignments given that the assumption of the left-to-right translation made for sentence alignment, is not valid at the bellow sentence level, or in other words, the order of the clauses in the source language is not the same in the target language. To handle cases of clause-alignments involving a number of clauses in the order of ten or more, we use a simulated annealing framework to approximate the optimal alignment. Simulated annealing (Metropolis et al., 1953), (Kirkpatrick et al. 1983), is a method for optimising functions depending on a large number of parameters. Annealing is a metallurgical term and the method is inspired by the controlled cooling of metals getting from the liquid to the solid state. The algorithm has been successfully applied for optimization purposes, including the approximate solution of TSP (Traveling Salesman Problem). This algorithm does not guarantee to find the best solution, but it may come up with a good approximation of it in non-exponential time. Processing starts with a random clause-alignment A. Initial temperature setting is $T=45$ and after each iteration it is reduced by 0.9. Each iteration is performed through 1000 steps. In each step, a random change in A is proposed and the cost function (negative logarithm of the clause-alignment probability) is computed. If the new alignment is better, the change is adopted, if not, it is adopted with probability $P = e^{-\frac{\Delta E}{T}}$, where ΔE is the change in the cost function. Once the loop is computed with no change in the configuration, or 10 iterations have been performed, the best alignment that has been found till that time is proposed.

Alignment Type	Initial Probability Estimation	Probability after Convergence
1-0	0.05	0.0001
1-1	0.8	0.6986
1-2	0.1	0.2465
2-2	0.05	0.0548

Table 1 : Initial and estimated probabilities

Results

The method has been applied to the corpus presented in section 2. A sample output of the method is displayed hereunder. Each table contains a source sentence, a target sentence and the set of proposed clause alignments (underlined alignments are wrong):

Alignment type:2-2, Dynamic Programming (DP)

[cl IF YOU HAVE NO CONSOLE cl] , [cl YOU MUST LOG IN FROM A REMOTE SYSTEM cl]
[cl ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ cl] [cl ΠΡΕΠΕΙ ΝΑ ΕΙΣΕΛΘΕΤΕ ΑΠΟ ΕΝΑ ΑΠΟΜΑΚΡΥΣΜΕΝΟ ΣΥΣΤΗΜΑ cl]
IF YOU HAVE NO CONSOLE <-> ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ *YOU MUST LOG IN FROM A REMOTE SYSTEM <-> ΠΡΕΠΕΙ ΝΑ ΕΙΣΕΛΘΕΤΕ ΑΠΟ ΕΝΑ ΑΠΟΜΑΚΡΥΣΜΕΝΟ ΣΥΣΤΗΜΑ*

Alignment type:3-3, DP

[cl THERE ARE SEVERAL REASONS cl] [cl THAT HP VUE MIGHT FAIL cl] [cl TO START cl]
[cl ΥΠΑΡΧΟΥΝ ΠΟΛΛΟΙ ΛΟΓΟΙ cl] [cl ΓΙΑ ΤΟΥΣ ΟΠΟΙΟΥΣ ΤΟ HP VUE ΜΠΟΡΕΙ ΝΑ ΑΠΟΤΥΧΕΙ cl] [cl ΝΑ ΞΕΚΙΝΗΣΕΙ cl]
THERE ARE SEVERAL REASONS <-> ΥΠΑΡΧΟΥΝ ΠΟΛΛΟΙ ΛΟΓΟΙ *THAT HP VUE MIGHT FAIL <-> ΓΙΑ ΤΟΥΣ ΟΠΟΙΟΥΣ ΤΟ HP VUE ΜΠΟΡΕΙ ΝΑ ΑΠΟΤΥΧΕΙ* *TO START <-> ΝΑ ΞΕΚΙΝΗΣΕΙ*

Alignment type:4-3, DP

[cl WHEN HP VUE FAILS cl] [cl TO BEHAVE cl] [cl AS EXPECTED cl] , [cl YOU SHOULD OPEN THE APPROPRIATE ERROR-MONITORING FILE cl]
[cl ΟΤΑΝ ΤΟ HP VUE ΑΠΟΤΥΓΧΑΝΕΙ cl] [cl ΝΑ ΣΥΜΠΕΡΙΦΕΡΘΕΙ ΚΑΤΑ ΤΟ ΑΝΑΜΕΝΟΜΕΝΟ cl] [cl ΘΑ ΠΡΕΠΕΙ ΝΑ ΑΝΟΙΞΕΤΕ ΤΟ ΚΑΤΑΛΛΗΛΟ ΑΡΧΕΙΟ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΣΦΑΛΜΑΤΩΝ cl]
WHEN HP VUE FAILS <-> ΟΤΑΝ ΤΟ HP VUE ΑΠΟΤΥΓΧΑΝΕΙ *TO BEHAVE AS EXPECTED <-> ΝΑ ΣΥΜΠΕΡΙΦΕΡΘΕΙ ΚΑΤΑ ΤΟ ΑΝΑΜΕΝΟΜΕΝΟ* *YOU SHOULD OPEN THE APPROPRIATE ERROR-MONITORING FILE <-> ΘΑ ΠΡΕΠΕΙ ΝΑ ΑΝΟΙΞΕΤΕ ΤΟ ΚΑΤΑΛΛΗΛΟ ΑΡΧΕΙΟ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΣΦΑΛΜΑΤΩΝ*

Alignment type:3-2, DP

[cl CREATING A SIMPLE ACTION cl] [cl COVERS cl] [cl HOW TO USE CREATEACTION cl]
[cl Η ΔΗΜΙΟΥΡΓΙΑ ΜΙΑΣ ΑΠΛΗΣ ΕΝΕΡΓΕΙΑΣ ΚΑΛΥΠΤΕΙ ΤΟ cl] [cl ΠΩΣ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΕΤΕ ΤΗ " CREATEACTION " cl]
CREATING A SIMPLE ACTION <-> Η ΔΗΜΙΟΥΡΓΙΑ ΜΙΑΣ ΑΠΛΗΣ ΕΝΕΡΓΕΙΑΣ ΚΑΛΥΠΤΕΙ ΤΟ *COVERS HOW TO USE CREATEACTION <-> ΠΩΣ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΕΤΕ ΤΗ " CREATEACTION "*

Alignment type:6-6, Simulated Annealing(SA)

[cl IF YOU PREVIOUSLY USED SOFTBENCH cl] [cl AND HAVE A PERSONAL <DIR>/HOMEDIRECTORY/.SOFTINIT </DIR> FILE cl] , [cl YOU MAY NEED cl] [cl TO REMOVE THE FILE cl] [cl OR EDIT IT cl] [cl TO INCLUDE THE HP VUE TOOLS cl]
[cl ΑΝ ΠΡΟΗΓΟΥΜΕΝΩΣ ΧΡΗΣΙΜΟΠΟΙΗΣΑΤΕ ΤΟ SOFTBENCH cl] [cl ΚΑΙ ΕΧΕΤΕ ΕΝΑ ΠΡΟΣΩΠΙΚΟ ΑΡΧΕΙΟ <DIR>/HOMEDIRECTORY/.SOFTINIT</DIR> cl] [cl ΜΠΟΡΕΙ ΝΑ ΧΡΕΙΑΣΤΕΙ cl] [cl ΝΑ ΑΦΑΙΡΕΣΕΤΕ ΤΟ ΑΡΧΕΙΟ cl] [cl Η ΝΑ ΤΟ ΤΡΟΠΟΠΟΙΗΣΕΤΕ cl] [cl ΩΣΤΕ ΝΑ ΠΕΡΙΛΑΜΒΑΝΕΙ ΤΑ ΕΡΓΑΛΕΙΑ HP VUE cl]
IF YOU PREVIOUSLY USED SOFTBENCH <-> ΑΝ ΠΡΟΗΓΟΥΜΕΝΩΣ ΧΡΗΣΙΜΟΠΟΙΗΣΑΤΕ ΤΟ SOFTBENCH *AND HAVE A PERSONAL <DIR>/HOMEDIRECTORY /.SOFTINIT</DIR> FILE <-> ΚΑΙ ΕΧΕΤΕ ΕΝΑ ΠΡΟΣΩΠΙΚΟ ΑΡΧΕΙΟ <DIR>/HOMEDIRECTORY/.SOFTINIT</DIR>* *YOU MAY NEED <-> ΜΠΟΡΕΙ ΝΑ ΧΡΕΙΑΣΤΕΙ* *TO REMOVE THE FILE <-> ΝΑ ΑΦΑΙΡΕΣΕΤΕ ΤΟ ΑΡΧΕΙΟ* *OR EDIT IT <-> Η ΝΑ ΤΟ ΤΡΟΠΟΠΟΙΗΣΕΤΕ* *TO INCLUDE THE HP VUE TOOLS <-> ΩΣΤΕ ΝΑ ΠΕΡΙΛΑΜΒΑΝΕΙ ΤΑ ΕΡΓΑΛΕΙΑ HP VUE*

The performance has been evaluated on a text portion containing ca. 250 sentences and overall precision of the output has been calculated to be 85.7%. If we exclude cases of misalignments due to errors in stages of processing preceding clause-alignment, we can calculate the precision of the last stage. In this case, precision is higher than 96%, so the error-rate introduced during clause-alignment is less than 4%. In addition to the low error-rate, clause-alignment corrects some of the errors caused by the previous stages, as it is mentioned in the next section.

Discussion

Given the incremental and engineering approach adopted, the results obtained so far are quite encouraging. The accuracy of the output lies around +85%, making the method quite reliable and suitable to be used in real world application systems.

Most of the errors were introduced by the first three primary processing stages, that is sentence-alignment, POS tagging and clause recognition. Major improvements in performance will certainly require further optimization of some or all of these stages along with any refinements to the statistical clause-alignment model used in the last stage. Regarding refinements to clause-alignment, there are several sources of information that could be readily taken into account. For example, pre-compiled bilingual dictionaries could be of help in order to establish reliable word associations in very short texts, which do not allow the safe estimation of the required word probabilities, while preference rules on clause types could be used to reduce search space, favoring alignments between certain clause types and penalising others. Future developments are believed to help improve accuracy and performance and broaden the coverage of the system in order to cover additional types of sub-sentence alignments. An interesting remark is that errors introduced by preceding stages are sometimes repaired by clause-alignment. For example, it may happen that a sentence is mistakenly chunked into clauses due to tagging or other errors. Then '1-2' and '2-2' clause-alignments may function in such a way that illegally separated sentence pieces are brought back together.

It is well understood that linguistic resources building is one of the important stumbling blocks in the localization/internationalization exercise. Methods approximating the automatic generation of such resources prove to be effective on a cost/time basis. Besides gains in speed and efficiency, the data driven approach improves consistency, which is an important requirement for systems

operating in a multilingual setting. By adopting a data driven approach and exploiting existing linguistic processing modules, the method produces textual parallel data of high resolution which can give a competitive advantage to multilingual processes and systems, such as semi-automatic lexicon builders, machine aided translation systems and retrieval of multilingual material.

References

Aho A., R. Sethi, and J. Ullman. 1986. Compilers, Principles, Techniques and Tools. Reading, Masschusets: Addison Wesley

Burnard, L. 1995. Users Reference Guide for the British National Corpus, British National Corpus Consortium Report, Oxford, England.

Boutsis, S., and S. Piperidis. 1996. Automatic Extraction of Bilingual Lexical Equivalences from Parallel Corpora. In Proc. Multilinguality in Software Industry /ECAI, 27-31.12 August, Budapest, Hungary.

Brown, P. 1988. A Statistical Approach to Language Translation. In Proc.12th International Conference on Computational Linguistics, vol. 1, 71-76. Budapest, Hungary.

Brown, P., J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roosin. 1990. A Statistical Approach to Machine Translation. Computational Linguistics. June: 79-85.

Brown P., J. Lai, and R. Mercer. 1991. Aligning Sentences in Parallel Corpora. In Proc. 29th Annual Meeting of the ACL, 169-176. 18-21 June, Berkley, Calif.

Dagan, I., A. Itai, and U. Schwall.1991. Two languages are more informative than one. In Proc. 29th Annual Meeting of the Association for Computational Linguistics, 130-137.18-21 June, Berkley, Calif.

Di Christo, P., S. Harie, C. De Loupy, N. Ide, and J. Veronis. 1995. Set of programs for segmentation and lexical look up, MULTEXT LRE 62-050 project Deliverable 2.2.1.

Elworthy, D. 1997. Tagger Suite User's Manual. Cambridge University Computer Laboratory Report, Cambridge, England.

Gale, W.A., and K.W. Church. 1991. A Program for Aligning Sentences in Parallel Corpora. In Proc. of the 29th Annual Meeting of the Association for Computational Linguistics, 177-184. 18-21 June, Berkley, Calif.

Gale, W.A., and K.W. Church. 1991b. Identifying word correspondences in parallel texts. Proceedings of the Fourth DARPA Speech and Natural Language Workshop, 152-157.

Kay, M., and M. Roescheisen. 1993. Text-translation Alignment. Computational Linguistics. March:121-142.

Kirkpatrick, S., C. Gelatt, and M.P. Vecchi. 1983. Optimisation by Simulated Annealing. Science Vol 220. pp. 671-680.

Kitamura, M., and Y. Matsumoto. 1995. A Machine Translation System based on Translation Rules Acquired from Parallel Texts. In Proc. Recent Advances in Natural Language Processing, 27-44. 14 - 16 September, Tzigov Chark, Bulgaria.

Kitamura, M., and Y. Matsumoto. 1996. Automatic Extraction of Word Sequence Correspondences in Parallel Corpora. In Proc. 4th Workshop on Very Large Corpora, 79-87. 4 August, Copenhagen, Denmark.

Kumano, A., and H. Hirakawa. 1994. Building an MT Dictionary from Parallel Texts Based on Linguistic and Statistical Information. In Proc. 15th International Conference on Computational Linguistics,76-81. 5-9 August, Kyoto, Japan.

Kupiec, J. 1993. An algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora. In Proc. 31st Annual Meeting of the Association for Computational Linguistics, 17-22. 22-26 June, Columbus, Ohio.

Leech, G. 1995. A brief users' guide to the grammatical tagging of the British National Corpus. British National Corpus Consortium Report, Oxford, England.

Matsumoto, Y., H.Ishimoto, T. Utsuro. 1993. Structural Matching of Parallel Texts. In Proc. 31st Annual Meeting of the Association for Computational Linguistics, 23-30. 22-26 June, Columbus, Ohio.

Metropolis, N., A. Rosenbluth, M. Teller, A. Teller, and E. Teller. 1953. Journal Chem. Phys. Vol. 21. Pp 1087.

Papageorgiou, H., L. Cranias, and S. Piperidis. 1994. Automatic Allignment in Parallel Corpora. In Proc. 32nd Annual Meeting of the Association for Computational Linguistics, 334-336. 27-30 June, Las Cruses, New Mexico.

Papageorgiou H. 1996. Part of Speech Disambiguation. In Hybrid Techniques for Bilingual Corpus Processing 63-83. PhD thesis, National Technical University of Athens, Greece

Piperidis S. 1995. Interactive Corpus-based Translation Drafting Tool. Aslib Proceedings. March: 83-92.

Piperidis, S., S. Boutsis, and I. Demiros. 1997. Automatic Translation Lexicon Generation. In Proc. Multilinguality in Software Industry /IJCAI. 25 August, Nagoya, Japan.

Simard, M., G. Foster, and P. Isabelle. 1992. Using cognates to align sentences in bilingual corpora. Proc. TMI-92. Montréal, Québec.

Smadja, F. 1992. How to compile a bilingual collocational lexicon automatically.In Proc. AAAI Workshop on Statistically -based NLP Techniques, 67-71. San Jose, California.

Smadja, F., K.R. McKeown, and V. Hatzivassiloglou. 1996. Translating Collocations for Bilingual Lexicons: A Statistical Approach. Computational Linguistics. March: 1-38.