An alignment architecture for Translation Memory bootstrapping

Ioannis Triantafyllou [1,2], Iason Demiros[1], Christos Malavazos [1,2], Stelios Piperidis[1,2]

[1] Institute for Language and Speech Processing, [2] National Technical University of Athens
Artemidos 6 & Epidavrou, 151 25, Athens, Greece
tel: +301 6875300, fax: +301 6854270
{yiannis, iason, christos, spip}@ilsp.gr

Abstract: In this paper, we describe an alignment environment that is fully embedded in the Tr•AID Integrated Translation Memory system. The alignment architecture consists of a text handler and a sentence recognizer, an ancillary bilingual lexicon builder, an anchor point disambiguator and finally a dynamic programming module. The tool has been extensively tested on real-world data and produces a success rate of ~95%. It is accompanied by an ergonomic editor helping the user post-editing its results. Future extensions on a more rigorous anchor points calculation are finally discussed.

Keywords: alignment, text handler, bilingual lexicon builder, anchor point disambiguator, dynamic programming.

1. Introduction

Real texts provide the current phenomena, usages and tendency of language in a particular space and time. Recent years have seen a surge of interest in bilingual and multilingual corpora, i.e. corpora composed of a source text along with translations of that text in different languages. One very useful organization of bilingual corpora requires that different versions of the same text be aligned. Given a text and its translation, an alignment is a segmentation of the two texts such that the nth segment of one text is the translation of the nth segment of the other. As a special case, empty segments are allowed and correspond to translator's omissions or additions.

The deployment of learning and matching techniques in the area of machine translation, first advocated in the early 80s by *Nagao (1984)* proposed as "Translation by Analogy" and the return to statistical methods in the early 90's [*Brown et al. (1993)*] have given rise to much discussion as to the architecture and constituency of modern machine translation systems. Bilingual text processing and in particular text alignment with the resulting exploitation of information extracted from thus derived examples created a new wave in machine translation (MT).

In this paper, we will describe a text alignment architecture for a computer-aided translation (CAT) platform implemented in the Tr•AID system. The system employs different levels of information and processing in an attempt to maximize past translation reuse as well as terminology and style consistency in the translation of specific types of text. The Tr•AID Aligner, fully embedded into the Tr•AID TM system, is a scalable bootstrapping tool used to construct sentence correspondences between parallel texts and to subsequently populate the Translation Memory. Accompanied by an ergonomic viewer and editor, the Tr•AID Aligner has been extensively tested on text types in 14 languages amounting to over a million words with exceptionally good results.

2. Background

Several different approaches have been proposed tackling the alignment problem at various levels. *Catizone et al. (1989)* introduced a technique to link regions of text according to the regularity of word co-occurrences across texts. *Brown et al. (1991)* described a method based on the number of words that sentences contain.

Gale & Church (1991) proposed a method that relies on a simple statistical model of character lengths. The model is based on the observation that lengths of corresponding sentences between two languages are highly correlated. Although the apparent efficacy of the Gale-Church algorithm is undeniable and validated on different pairs of languages (English-German- French-Czech-Italian), it seems to be awkward when handling complex alignments.

Given the availability in electronic form of texts translated into many languages, an application of potential interest is the automatic extraction of word equivalencies from these texts. *Kay & Roscheisen (1991)* have presented an algorithm for aligning bilingual texts on the basis of internal evidence only. This algorithm can be used to produce both sentence alignments and word alignments.

Simard et al. (1992) argues that a small amount of linguistic information is necessary in order to overcome the inherited weaknesses of the Gale-Church method. He proposed using cognates, which are pairs of tokens across different languages that share "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.

3. System Architecture

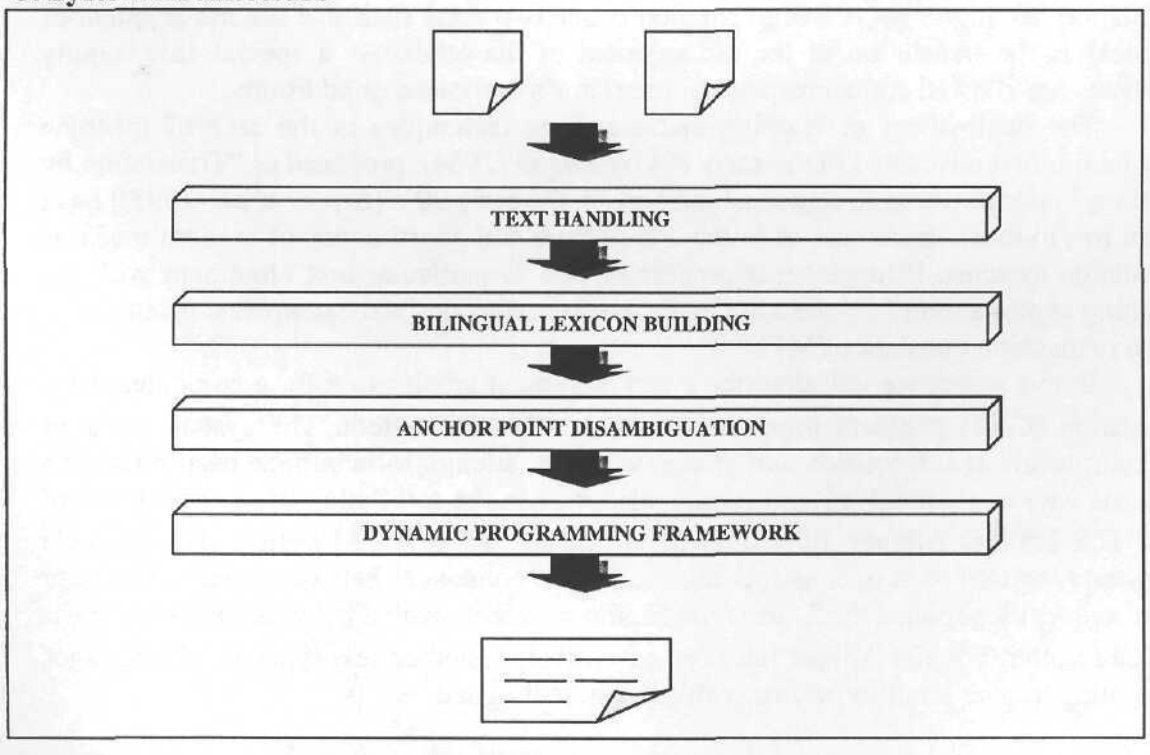

Figure 1: TrAID Aligner Architecture

Figure 1 displays the alignment architecture where all the individual components are presented within the overall framework. The parallel texts are preprocessed by the text handler and then a set of corresponding single or multi-word units is constructed. The set of corresponding units, or points, supplemented with sentence boundary information and filtered by a disambiguation module, expresses sentence correspondence (anchor points). A Dynamic Programming (DP) alignment between anchor points completes the alignment process. A detailed description of each individual system component is presented in the following sections.

3.1 Text Handling

Recognizing and labelling surface phenomena in the text is a necessary prerequisite for most Natural Language Processing (NLP) systems. At this stage, texts are rendered into an internal representation that facilitates further processing. Basic text handling is performed by a MULTEXT-like tokenizer [*Di Christo et al. (1995)*], that identifies word boundaries, sentence boundaries, abbreviations, digits, and simple dates. Following common practice, the tokenizer makes use of a regular-expression based definition of words, coupled with downstream precompiled lists for abbreviations in many languages and simple heuristics. This proves to be quite successful in effectively recognizing sentences and words, with accuracy up to 96%. The text Handler is responsible for transforming the parallel texts from the original form in which they are found into a form suitable for the manipulation required by the application. Although its role is often considered as trivial, in fact it is crucial for subsequent steps that heavily rely on correct word boundary identification and sentence recognition. The lexical resources of the handler are constantly enriched with abbreviations, compounding lists, dates, number and enumeration formats for new languages.

A certain level of interactivity with the user has been introduced in the Tr•AID system by indirect manipulation of pre-fixed sentence splitting patterns such as a new line character followed by a word starting with an uppercase character. The text handler has been ported to the Unicode standard and the number of treated and tested languages has reached 14: English, French, German, Italian, Spanish, Italian, Portuguese, Swedish, Dutch, Finnish, Danish, Greek, Bulgarian, Russian. We are currently working on several other Balkan and N.I.S. languages which in the near future will be part of the family of the official Community languages.

3.2 Bilingual Lexicon Building

3.2.1 Lexical correspondences

The process of finding correspondences of arbitrary length between parallel word sequences is based on the *Kitamura and Matsutomo (1997)* algorithm although they require pre-aligned and morphologically analyzed parallel corpora in English and Japanese. The steps of our lexicon building process are outlined below:

1. All word sequences of length up to three are candidates as lexicon entries.

2. Functional words are excluded from the candidate list. Stop lists for 10 languages have been made available and they are used for this purpose. If such a resource is unavailable

for both languages, noise from functional word lexicon entries is propagated to the next steps and leads to a certain degradation of the aligner performance (for instance when aligning Finnish to Danish).

3. A modification of the Dice coefficient is calculated as the similarity value between two word sequences:

$$sim(w_A, w_B) = (\log_2 f_{AB}) \frac{2 f_{AB}}{f_A + f_B}$$

where w_A and w_B are word sequences in the parallel texts A and B, f_{AB} is the co-occurrence frequency of w_A and w_B and f_A, f_B are the frequencies of w_A and w_B respectively. Only the pairs with similarity value greater than a threshold $\log_2 f_{\min}$ are considered in this step, where $f_{\min}$ is the half of the highest number of occurences of the candidate word sequences. Also, only the pairs with maximum normalized deviation less than $O(\sqrt{n})$, threshold introduced in *Kay & Roscheisen (1991)*, where n is the mean value of the number of sentences contained in the parallel texts, are considered for co-occurrence. In order to calculate f_{AB}, we consider the non-overlapping pairs that are closer to the normalized main diagonal of the parallel texts.

4. The most plausible correspondences between word sequences are identified by the following process:
For a word sequence w_A in text A let $\{w_{Bk}\}$ be the set of all word sequences in its target text B such that for each k, $sim(w_A, w_{Bk}) \geq \log_2 f_{\min}$. The set is the candidate set of w_A. For each word sequence in $\{w_{Bk}\}$ we construct its candidate set, too. Suppose that w_{Bj} is the candidate sequence in $\{w_{Bk}\}$ yielding the higher similarity score with w_A. If w_{Bj} again selects w_A as the candidate with the highest score, the pair (w_A, w_{Bj}) is a valid translation pair.

5. The threshold $f_{\min}$ is lowered by dividing by 2 until it reaches (or drops below) 10 and then it is lowered by 1 until a predefined value is reached.

3.2.2 Long sentence filtering

Apart from the above mentioned lexical translation pairs, long sentence correspondences are entered as a virtual item in the bilingual lexicon, following the observation that their distributions in the parallel texts are quite similar. From the length vectors of the sentences in the parallel texts, we find the maximum likelihood estimates of the parameters *(a,b)* of a gamma distribution that we fit to the vectors. Finally we retain only those sentences whose probability of being an observation from a gamma distribution with parameters *(a,b)* will be less than 1%. An experiment with the tail of a Poisson distribution yielded similar results. This test is mostly used to confirm or reject anchor points disambiguated as described below.

3.3 Anchor Point Disambiguation

In this step, the correspondences between word sequences are used to establish a mapping between sentences of the parallel texts. Points of correspondence, which are often referred to as anchor points, are generated from a process that applies to each pair of word sequences (w_A, w_B). Let $(s(w_A), s(w_B))$ be the word-sentence index pair for the word sequence pair (w_A, w_B). It is a pair of arrays showing the index number of the sentences in which the word sequences (w_A, w_B) occur. The sentences $s_i(w_A)$ and $s_j(w_B)$ are accepted as a true point of correspondence if their normalized distance from $s_{i+1}(w_A), s_{i-1}(w_A)$ and $s_{j+1}(w_B), s_{j-1}(w_B)$ respectively is greater than $O(\sqrt{n})$, where n is again the mean value of the number of sentences contained in the parallel texts . If the distance in one of the two arrays is less than $O(\sqrt{n})$, $s_i(w_A)$ and $s_j(w_B)$ are rejected. If both are less than $O(\sqrt{n})$, we apply a supplementary test on the deviations of the mutual normalized differences in order to capture the case where, although the positions are very close to each other and by consequence quite ambiguous, their distributions are very similar and as such they will be accepted as true points of correspondence.

3.4 Dynamic Programming

The sentence-level alignment of regions between anchor points is based on the simple but effective statistical model of character lengths introduced by *Gale & Church (1991)*. The model makes use of the fact that longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter sentences. A probabilistic score is assigned to each pair of proposed sentence pairs, based on the ratio of lengths of the two sentences (in number of characters). The score is computed by integrating a normal distribution. This probabilistic score is used in a dynamic programming framework in order to find the Viterbi alignment of sentences.

4. Description of an experiment

In the following we present an experiment illustrating the alignment process. For the purposes of this example, we have used parallel texts from the legal domain, in English and Spanish. The settings of the experiment were:

Number of English sentences: 763
Number of Spanish sentences: 769
Normalization factor: 1.00786
$f_{\min} = 256 / 2$

word sequence correspondences: 67
Error (linguistically incorrect): 9.1%
Error: 3.6%

A sample of the extracted word sequences appears hereafter:

[Threshold] ---> SourceWord <--> TargetWord
<SrcOccur1,SrcOccur2,...> <TrgOccur1,TrgOccur2,...>

[006] ---> REPUBLIC <--> REPUBLICA
<739,739,742,742,748,749,751,753,756,757,758> <740,740,743,743,749,750,752,754,757,758,759>
[006] ---> SERIOUS <--> GRAVES
<27,32,103,200,309,330,707> <27,32,103,199,306,327,703,705>
[006] ---> RELATED <--> CONEXOS
<111,112,198,217,218,237,729,730> <111,112,197,216,216,234,725>
[005] ---> COMPUTERIZED&SYSTEM <--> RECOGIDA&DATOS
<45,125,147,177,178,182> <45,125,147,175,176,180>
[005] ---> WORK&FILES <--> FICHEROS&TRABAJO
<50,180,234,372,383,387> <50,178,231,369,380,384>
[005] ---> EUROPEAN&POLICE <--> OFICINA
<8,10,14,23,30,99> <9,11,15,23,30,99>
[004] ---> MEMBER&STATES&EUROPEAN <--> MIEMBROS&UNION&EUROPEA
<25,28,99,256,732> <25,28,99,253,730>
[004] ---> RIGHT&ACCESS <--> DERECHO&ACCESO
<49,61,220,360,361> <49,61,219,357,358>
[004] ---> 21 <--> 21
<63,193,195,400,519> <63,192,194,396,514>
[003] ---> OBLIGATION&DISCRETION <--> OBLIGACION&RESERVA
<75,568,570,572,584> <75,567,578,579>
[003] ---> UNAUTHORIZED&INCORRECT <--> LITIGIOS
<82,519,642,646> <84,514,649,651>
[003] ---> JULY <--> JULIO
<1,26,703,731> <1,26,699,728>

The number in brackets is the frequency that the pair following was produced. In the next line appear the word-sentence index vectors for each pair. It is noticeable that we have extracted a few linguistically incorrect pairs such as EUROPEAN&POLICE <--> OFICINA that yield perfectly correct anchor points in the next step. Also, there are incorrect pairs like UNAUTHORIZED&INCORRECT <--> LITIGIOS that produce incorrect anchor points and false alignments. For the pair UNAUTHORIZED&INCORRECT <--> LITIGIOS, for instance, the anchor point 82-84 is incorrect, the point 519-514 is correct and the last two points were rejected by the disambiguator.

Anchor points generated: 330
Anchor points rejected by disambiguation: 96
Anchor points remaining: 234
Incorrect anchor points: 5

Below we present a snapshot from the disambiguation procedure on the pair RELATED-CONEXOS

File001: (RELATED-CONEXOS) 0111-0111
File001: (RELATED-CONEXOS) 0112-0112
File001: (RELATED-CONEXOS) 0198-0197
File001: (RELATED-CONEXOS) 0217-0216****REJECTED****[next-src&trg also matched---dist<=28]
File001: (RELATED-CONEXOS) 0218-0216****REJECTED****[prev-src&trg also matched---dist<=28]
File001: (RELATED-CONEXOS) 0237-0234
File001: (RELATED-CONEXOS) 0729-0725****REJECTED****[next-src&trg also matched---dist<=28]

Total number of alignments after the dynamic programming module: 747
Correct alignments: 35
Error: 4.7%

5. Evaluation and ongoing work

The Tr•AID Translation Memory environment was designed to enhance the translator's work by making use of previously translated and stored text. The user builds his/her own translation memories through the Aligner tool that is provided as a standard component of the environment. He/she can visualize, inspect and edit the alignments that are produced and finally bootstrap his/her translation work by loading the alignments into the database. The pressure from customer requirements lead to extensive testing and evaluation on different kinds of corpora and continuous work on the quality and robustness of the tool. The variety of applications and text types has forced us to increase speed, create APIs and improve the software reusability.

Dealing with our users' documents has proved much different than aligning the Hansards or the Celex directives, the EU's documentation system on EU law. We have aligned product manuals where the lexical unit and sentence recognition error rate was very high – almost 50% - due to the peculiar format of the input documents. Obviously the dynamic programming component of the Aligner failed in such cases to produce a correct alignment. For other languages such as Bulgarian, lack of abbreviation lists leads to errors in sentence recognition, errors that affect the aligner success. When dealing with texts that do not exhibit strange phenomena, sentence recognition success is in the range 95-97% and aligner success 93-95%. A powerful and user-friendly misalignment correction environment helps the user handle the remaining errors and produce an error-free aligner that is stored in the Translation Memory database.

We are currently working on exploiting the full spectrum of information provided by the text handler such as dates, numbers and enumeration patterns to further improve the aligner output. We also set some degrees of anchorness so that the user can spot the difficult regions where mistakes might have a higher probability of occurrence. Research on feature patterns of sentences as well as on evolutionary programming of points of correspondence alignment is being planned but such techniques will be incorporated in our system only if the gain proves to be significant.

6. References

Nagao (1984), M. Nagao, *A framework of a mechanical translation between Japanese and English by analogy principle.* Artificial and Human Intelligence, ed. Elithorn A. and Banerji R., North-Holland, pp 173-180, 1984.

Brown et al. (1993), P. F Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer, *The Mathematics of Statistical Machine Translation: Parameter Estimation*, Computational Linguistics, June 1993.

Catizone et al. (1989), R. Catizone, G. Russell, S. Warwick, *Deriving translation data from bilingual texts,* Proc. of the First Lexical Acquisition Workshop, Detroit 1989

Brown et al. (1991), P. F Brown, J. C. Lai, R. L. Mercer, *Aligning Sentences in Parallel Corpora,* Proc. of the 29th Annual Meeting of the ACL, pp 169-176, 1991.

Gale & Church (1991), W. A. Gale and K. W. Church *A Program for Aligning Sentences in Bilingual Corpora.* Proc. of the 29th Annual Meeting of the ACL., pp 177-184, 1991.

Kay & Roscheisen (1991), M. Kay, M. Roscheisen, *Text-Translation Alignment,* Computational Linguistics Vol. 19, No 1, 1991.
Simard et al. (1992), M. Simard, G. Foster and P. Isabelle, *Using cognates to align sentences in bilingual corpora*, Proc. of TMI, 1992.
Papageorgiou et al. (1994), H. Papageorgiou, L. Cranias and S. Piperidis, *Automatic alignment in parallel corpora*, Proc. of the 32nd Annual Meeting of the ACL, 1994.
Di Christo et al. (1995) Set of programs for segmentation and lexical look up, MULTEXT LRE 62-050 project Deliverable 2.2.1 (1995)
Kitamura and Matsumoto (1997), Automatic extraction of word sequence correspondences in parallel corpora, 4th Workshop of VLC, 1997.