TR•AID: A Memory-based Translation Aid Framework

Stelios Piperidis, Christos Malavazos, Ioannis Triantafyllou

Institute for Language and Speech Processing
Language Technology Applications Department
Epidavrou & Artemidos 15125 Marousi, Greece
spip, christos, yiannis@ilsp.gr

Abstract

This paper describes **TR•AID,** a multi-level architecture for a computer-aided translation (CAT) system platform. 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. Such tools have come in the bibliography under the term Translation Memory (TM) tools.

Keywords: Machine Translation/**MT**, Memory Based translation/**MBT**, Example Based Machine Translation/**EBMT**, Computer Aided Translation/**CAT**, Term Spotting.

1 Introduction

The deployment of learning and matching techniques in the area of machine translation, first advocated in the early 80s (Nagao 84) proposed as *"Translation by Analogy"* and the return of statistical methods in the early 90's (Brown et al. 93) 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 has turned into a new wave in MT.

Traditional Rule-Based Machine Translation (**RBMT**) systems suffer from tractability, adaptability as well as quality and performance problems. Example-based Machine Translation (**EBMT**) also known as Memory-based Machine Translation (**MBMT**) has attempted to provide alternative ways to overcome the knowledge acquisition bottleneck, yielding promising results.

1.1 Background

Translation work is often characterised by three conflicting parameters: repetition, demand on efficiency as well as high demand on quality, especially in terms of consistency. This is particularly true for translation of technical and administrative documentation, becoming more evident in the case of law documents and product documentation where text repetition may reach a rate of 70% and sometimes higher.

TR•AID aims at providing a computational framework, in more practical terms a toolbox that will:

- rid translators of the repetitive part of their work by reusing existing human translations and learning from them
- enhance quality and consistency of translation by being able to integrate ancillary translation tools.

Appropriate storage of pairs of source language (SL) and target language (TL) blocks of text and provision of means for retrieval of applicable solutions and means for post-editing them would increase the productivity of a translator and at the same time improve the quality and consistency of the translation (Freibott 92) (Ishida 94).

The key issues of the approach revolve around four major axes:

- "automatic" alignment of parallel texts, i.e. establishment of correspondences between units of parallel texts
- organisation of multilingual parallel corpora, i.e. texts in different languages, one being the translation of the other, allowing for efficient storage and retrieval of translation examples as well as terminological data.
- sophisticated text matching techniques for fast retrieval of most appropriate translation templates
- sophisticated "term conflation" techniques for term spotting and translation.

Alternative techniques have been examined under the proposed architecture for each individual task. The most practical as well as cost-effective solutions have been adopted and integrated towards the development of the **TR•AID** (**Translation Aid**) **system.**

2 System Architecture

2.1 Overview

Figure 1 displays **TR•AID's** architecture where all the individual components are presented within the overall framework. A detailed description of each individual system component will be provided in the following sections.

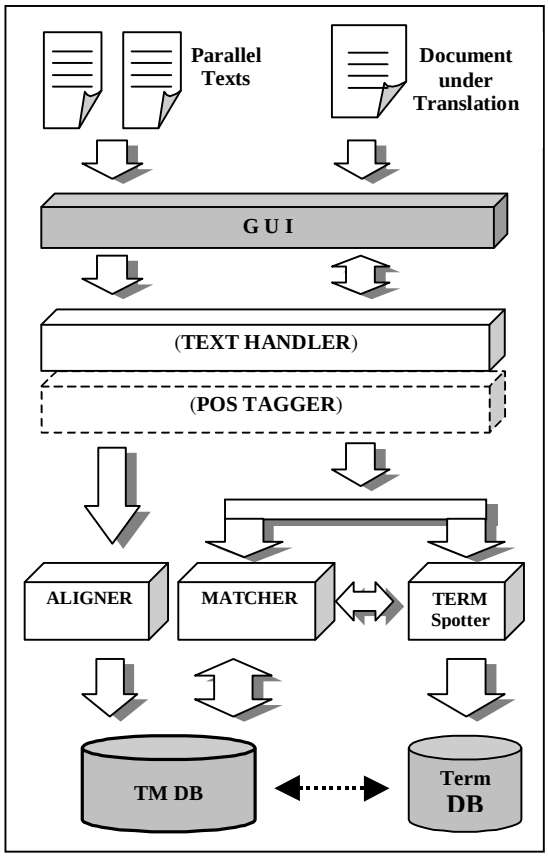

Figure 1: TrAID Architecture

2.2 Text Handling

In order to be able to make full use of parallel corpora, the corpora have to be rendered in an appropriate form. To this end, corpora have to be normalised and handled prior to alignment. **Normalisation** consists in extraction from the multilingual corpus body of all those sections or information that cannot be exploitable for text translation purposes.

Text handling can be seen as a sophisticated interface between input text streams and various text manipulation modules. At the stage of analysis, the text handler has the responsibility of transforming a text from the original form in which it is found into a form suitable for the manipulation required by the application; at the stage of synthesis, it is responsible for the reverse process, i.e. for converting the output text from the form used by the application into a form equivalent to that of the input text. The main operations usually associated with the text handler include:

- analysis of the **format** of the physical appearance of the input text (as evidenced by the word-processing and/or typesetting commands, such as bold and italic characters, indentation, etc.) and mapping of these into a standardised mark-up language or a canonical form recognised by the application
- identification of **textual units** at the level of paragraphs and sentences
- identification of **extra-linguistic** elements, such as dates, abbreviations, acronyms, list enumerators, numbers, etc.
- at the stage of synthesis, conversion of the output of the application into the same format recognised at the stage of analysis; e.g. italicised characters, centred phrases, etc. must be given to the user in their original form.

In the last few years, we have seen notable work on tokenization and sentence segmentation. (Grefenstette & Tapanainen 94) apply regular expression grammars with abbreviation lists and improve sentence recognition by adding increasing levels of linguistic sophistication. (Palmer & Hearst 94) have developed an efficient, trainable algorithm that uses a lexicon with part-of-speech probabilities and a feed-forward neural network. (Chanod & Tapanainen 96) propose a finite-state automaton for simple tokens and a lexical transducer that encodes a wide variety of multiword expressions. (Reynar & Ratnaparkhi 97) propose a solution based on a maximum entropy model which requires a few hints about what information to use and a corpus annotated with sentence boundaries.

Following common practice, a multilevel architecture is proposed, consisting of regular expression definition of words, coupled with precompiled common abbreviation lists for the treated language and simple heuristics for

distinguishing between these abbreviations or other evident abbreviation. Scalability has been considered as a crucial factor during the design and implementation.

Depending on the availability of corpus linguistic annotators in the languages represented in the multilingual corpus, the corpus is **lemmatised** and **tagged** for grammatical category (part of speech, pos). Possible unresolved ambiguities stemming from multiple possible lemma and tag assignments are appropriately stored in the memory.

2.3 Text Alignment

One crucial factor in establishing an alignment methodology, is the nature of the "text-units" involved. Deciding about the "text-units", that is determining whether the search is for matches at sentence or sub-sentence level, mainly concerns the best match retrieval component. Sentences, however, constitute the sole mostly unambiguous text unit and on this ground sentence level has been chosen for text alignment within the **TR•AID** framework.

Several different approaches have been proposed tackling the alignment problem at various levels. Catizone's technique (Catizone et al. 89) was to link regions of text according to the regularity of word co-occurrences across texts. (Brown et al. 91) described a method based on the number of words that sentences contain. Moreover, certain anchor points and paragraph markers are also considered. The method has been applied to the Hansard Corpus and has achieved an accuracy between 96%-97%.

(Gale & Church 91) 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 91) 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. 92) 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 of different languages which share "obvious" phonological or orthographic and semantic properties, since these are likely to be used as mutual translations.

(Papageorgiou et al. 94), 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 tags. The results are then fed into a dynamic programming framework that computes the optimum alignment of text units.

The proposed alignment scheme consists of a multi-level architecture employing as a core engine the Gale-Church mechanism. Special effort has been made to improve the performance of the former mechanism by locating candidate anchor points based only on internal evidence. Candidate word alignments are computed based on individual word, bi-word and tri-word distribution. Based on word alignment information, the most reliable sentence pairs are extracted. These are used subsequently, as boundaries within which the core engine will run thus providing better results. Alternatively, significant improvement can be made at this point by employing possibly available bilingual lexica.

2.4 Underlying Database

The complexity inherent in the translation processes within a typical EBMT framework necessitates the existence of well-defined powerful resources. Optimal utilisation of different levels of available resources as well as the need to provide real time responses, calls for an efficient as well as complete database architecture.

We define as meta-data the distinguishable objects present in the translation memory application, derived from the original raw text through the text pre-processing (annotation) and alignment process, as previously described. The

proposed architecture apart from the plain storage of monolingual corpus meta-data will also need to account for the appropriate storage of bilingual meta-data which will render the monolingual corpora as parallel aligned corpora. The derivation of supplementary bilingual meta-data, such as multi-word units or fixed phrases cross language associations, should also be possible to be later accommodated under the same framework.

The meta-data physically stored in our DB schema have been further decomposed into the following logical entities:

- **Words**: all wordforms appearing in the texts
- **Lemmas**: all the lemma forms from which any wordform in the text can be derived
- **Tags**: POS tags (grammatical categories) of each word in the text
- **Sentences**: basic structural units
- **Documents**: the files comprising the corpus
- **Corpus**: collection of the above
- **Translation Memory**: folder associated with a particular subject domain and possibly a particular user. It comprises all of the above and can be conceived as a super-entity.

Database administration is handled by a number of mechanisms especially designed for this purpose. The user is provided with batch as well as interactive procedures for inserting new translation examples into the DB and for managing DB modules (creating,deleting, loading, updating).

2.5 Text Matching

In establishing a mechanism for the best match retrieval two crucial tasks are identified:
(i) determining whether the search is for matches at sentence or sub-sentence level, that is determining the "**text unit**", and
(ii) the definition of the **metric** of similarity between two text units.

Sentences constitute the basic text unit in the translation process. This is because, not only are sentence boundaries unambiguous, but also translation proposals at sentence level is what a translator is usually looking for. Sentences can, however, be quite long. And the longer they are, the less possible it is that they will have a perfect match in the translation archive, and the less flexible the EBMT system will be.

On the other hand, if the text unit is the sub-sentence, it is likely that the resulting translation of the whole sentence will be of low quality, due to boundary friction (Sato & Nagao 90) and incorrect chunking. In practice, EBMT systems that operate at sub-sentence level involve the dynamic derivation of the optimum length of segments of the input sentence by analysing the available parallel corpora. This requires a procedure for determining the best "cover" of an input text by segments of sentences contained in the database (Nirenburg et al. 93). It is assumed that the translation of the segments of the database that cover the input sentence is known. What is needed, therefore, is a procedure for aligning parallel texts at sub-sentence level (Kaji et al. 92), (Sadler & Vendelmans 90). If sub-sentence alignment is available, the approach is fully automated but is quite vulnerable to the problem of low quality, as well as to translational ambiguity problems when the produced segments are rather small. Despite the fact that almost all running EBMT systems employ the sentence as the text unit, it is believed that the potential of EBMT lies on the exploitation of fragments of text smaller than sentences and the combination of such fragments to produce the translation of whole sentences (Sato & Nagao 90).

Turning to the definition of the metric of similarity, the requirement is usually twofold. The similarity metric applied to two sentences should indicate how similar the compared sentences are, and perhaps the parts of the two sentences that contributed to the similarity score. The latter could be just a useful indication to the translator using the EBMT system, or a crucial functional factor of the system.

The similarity metrics reported in the literature can be characterised depending on the text patterns they are applied on. So, the word-based metrics compare individual words of the two sentences in terms of their morphological paradigms, synonyms, hyperonyms, hyponyms, antonyms, pos tags (Nirenburg et al. 93) or use a semantic distance d ($0 \leq d \leq 1$) which is determined by the Most Specific Common Abstraction (MSCA) obtained from a thesaurus abstraction hierarchy (Sumita & Iida 91). Then, a similarity metric is devised, which reflects the similarity of two sentences, by combining the individual contributions towards similarity stemming from word comparisons.

The word-based metrics are the most popular, but other approaches include syntax-rule driven metrics (Sumita & Tsutsumi 88), character-based metrics (Sato 92) as well as some hybrids (Furuse & Iida 92) (Cranias et al. 94). The character-based metric has been applied to Japanese, taking advantage of certain characteristics of Japanese. The syntax-rule driven metrics try to capture similarity of two sentences at the syntax level. This seems very promising, since similarity at the syntax level, perhaps coupled by lexical similarity in a hybrid configuration, would be the best an EBMT system could offer as a translation proposal. The real time feasibility of such a system is, however, questionable, since it involves the complex task of syntactic analysis.

The third key issue of EBMT, that is exploiting the retrieved translation example, is usually dealt with by integrating into the system conventional MT techniques (Kaji et al. 92), (Sumita & Iida 91). Simple modifications of the translation proposal, such as word substitution, would also be possible, provided that alignment of the translation archive at word level or domain specific lexica are available.

The core of the **TR•AID** system is its text matching tool. Having rendered the corpus in the appropriate form (handled, aligned), the matching tool can search for database sentences that are identical or only similar to an input sentence and in addition retrieve the equivalent translation.

The matching mechanism consists of two processes:
(i) the perfect match process by which the system finds a database sentence (and its translation) in the Translation Memory which is identical to the input sentence, and
(ii) extraction of candidate sentences and the fuzzy match process. The fuzzy match process aims at extracting from the TM a number of sentences and their translations which resemble the given input sentence above a certain minimum degree (percentage), specified by the user.

2.5.1 Perfect (Full) Match Mechanism

This process consists in searching for perfect (full) matches between the input and the database sentences. In doing so, it uses statistical information in order to quickly and efficiently locate a small set of candidate database sentences within which the existing perfect matches will reside, if any. Furthermore, in order to cope with minor differences and overcome to some extent the flexibility problem, the perfect match process does not take into account extra-linguistic tokens so that linguistically perfect matches are not missed due to this kind of variations.

If no perfect match is found, the matching tool searches for database sentences that are similar to the input, i.e. for fuzzy matches.

2.5.2 Fuzzy Match Mechanism

The aim of the second phase of the matching mechanism is to find a sentence or a set of sentences in TM which are as similar as possible to the input sentence. The approach adopted to text matching is based on computations of common elements between sentences as well as computation of consecutive elements in them. The level at which computations of common elements are performed can vary between wordform level and lemma-tag level depending on the available resources, i.e. computations are either based on wordforms and their respective position in the compared sentences or on lemma-tag tuples of each word in the compared sentences as well as their respective positions in them.

For efficiency reasons, this phase of the matching process is separated into two stages:
(i) extraction of a small set of candidate sentences.
(ii) fuzzy match procedure.

The aim of the first stage is to extract a small list of candidate sentences bearing some common characteristics with the input sentence. This stage is used in order to reduce the search space and to improve the system's response time. Sentence length, individual words or word sequences of variable length have been alternatively studied and used in this stage.

The aim of the fuzzy match procedure is to extract the best match sentence out of the previous set of candidate sentences. Each sentence is encoded into a vector based on the elements it contains. Then a Dynamic Programming pattern matching technique (Ney 84) takes place producing a similarity score for each sentence based on the common and contiguous segments as well as the length of the

sentences under comparison. The common as well as the different elements of the two sentences that contributed to this score are located and presented to the user so that he/she adapts efficiently the suggested translation. In the simplest case an element corresponds to a wordform. The procedure can be expanded to encapsulate surface linguistic information, in which case, the element is a combination of a word and a lemma (and/or a pos tag).

Exemplary cases of fuzzy matches computed by the matching tool include: Sa, Sb, Sc, Sd stand for segments of sentences extending over a number of words identified in the input sentence (IS) and database sentences (DS).

IS : Sa Sb	**IS** : Sa Sc	**IS** : Sa Sb ***Sc***
DS : Sa Sb ***Sc***	**DS** : Sa ***Sb*** Sc	**DS** : Sa Sb ***Sd***
IS : Sa Sb ***Sc***	**IS** : Sa ***Sb*** Sc	**IS** : Sa ***Sb Sc***
DS : Sa Sb	**DS** : Sa Sc	**DS** : Sa ***Sc Sb***

Figure 2: Possible fuzzily matching segments

Several other experiments have been made in order to finally decide on the proposed similarity metric as well as the particular matching algorithm to be used. Interesting results were observed through the use of an "enhanced" string edit distance algorithm. This particular algorithm is based on a dynamic programming framework and on the same sentence representation scheme as the previous one and aims at estimating the minimum transformation cost between two sentences. The algorithm computes the minimum number of required editing actions (insertions, deletions, substitutions, movements and transpositions) in order to transform one sentence into another through an inverse backtracking procedure. The final similarity score is computed by assigning appropriate weights to these actions. Even though this method achieves a more thorough comparison between sentences it is still under question whether this will finally constitute a more cost-effective solution.

In cases where fuzzy matches accepted by the user are found, the user is asked to render in the target language those parts of the SL sentence that have not matched. The new emerging pair of translation units is then stored in the translation memory database for future use. In cases where no match can be found, including cases where matches exist but their score is below the user's desired threshold, the user is asked to provide the translation of the IS which is again subsequently stored in the TM database. Thus, the translation memory system starts learning new translation pairs in an interactive mode.

2.6 Term Spotting and Translation

Term spotting and translation has been included within the overall **TR•AID** framework as an intermediate step towards a full document translation process. This tool spots candidate terms and replaces them with their translation equivalents (if any) in the desired (native) language. In both steps the system uses a multilingual terminological database, mainly to identify a term and then to get its translation. The underlying DB schema emphasises on the efficient storage of monolingual as well as bilingual information allowing for fast retrieval.

The system aims at capturing morphological variations of terms located in the database, through a "term conflation" process (Frakes 84). Term conflation is being performed at search time allowing for full form information to be stored in the DB. For efficiency reasons, the term spotting process is performed in two subsequent phases. The first phase aims at reducing the search space thus improving the performance of the system in terms of required memory recourses as well as response time. During this, the system extracts a small set of candidate terms based on statistical information. Subsequently, during the second phase a more elaborate procedure takes place, where the systems ranks the located terms producing a complete term "short-list" for each candidate term of the input text. The scoring mechanism is based on a dynamic programming framework, especially designed to assign higher scores to morphological variations of the same root form. The system can easily detect single as well as multiword terms and also exclude functional words from the matching process, if these are available.

An interesting aspect of the term substitution task that is currently being investigated is how this could be fully integrated within the sentence matching process that is, to actually use term existence information during sentence matching and translation.

3 Concluding Remarks

The real added value of a translation related software is in its ability to enhance the efficiency of the translation task by cutting down cost and time while retaining quality of a purely human generated translation. Fully automatic machine translation is not yet feasible. The goal should be to develop a system that optimally combines different levels of sophistication and resources and which will be adaptable to different languages and domains.

Acknowledgements

This work has been conducted in part in the framework of the LE-I 2238 AVENTINUS project funded by the Commission of the European Union.

References

(Brown et al. 91), 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.

(Brown et al. 93) 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. 89) R. Catizone, G. Russell, S. Warwick, *Deriving translation data from bilingual texts,* Proc. of the First Lexical Acquisition Workshop, Detroit 1989

(Chanod & Tapanainen 96) J. P. Chanod and P. Tapanainen. *A non-deterministic tokenizer for finite-state parsing*, Proceedings of the ECAI 96 Workshop, 1996.

(Cranias et al. 94) L. Cranias, H. Papageorgiou and S. Piperidis, *A matching technique in Example-Based Machine Translation*, Proc. of Coling, pp 100-105.

(Frakes 84) W. B. Frakes *Term Conflation for Information Retrieval.* Research and Development in Information Retrieval, New York: Cambridge University Press, 1984.

(Freibott 92) G.P. Freibott, *Computer Aided Translation in an Integrated Document Production Process:Tools and Applications*, Translating and the Computer 14, pp 45-66, 1992.

(Furuse & Iida 92) O. Furuse and H. Iida, *Cooperation between Transfer and Analysis in Example-Based Framework*. Proc. Coling, pp 645-651, 1992.

(Gale & Church 91) 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.

(Grefenstette & Tapanainen 94) G. Grefenstette and P. Tapanainen *What is a word, What is a sentence? Problems of tokenization*, COMPLEX 94.

(Ishida 94) R. Ishida, (1994), *Future translation workbenches: some essential requirements*, Aslib Proceedings, vol.46, no. 6, pp 163-170, June 1994.

(Kaji et al. 92) H. Kaji, Y. Kida and Y. Morimoto, *Learning Translation Templates from Bilingual Text*. Proc. Coling., pp 672-678, 1992.

(Kay & Roscheisen 91) M. Kay, M. Roscheisen, *Text-Translation Alignment,* Computational Linguistics Vol. 19, No 1, 1991.

(Nagao 84) 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.

(Ney 84) H. Ney, *The use of a One-stage Dynamic Programming Algorithm for Connected Word Recognition*, IEEE vol. ASSp-32, No 2, 1984.

(Nirenburg et al. 93) S. Nirenburg, C. Domashnev D. J. Grannes. *Two Approaches to Matching in Example-Based Machine Translation*. Proc. of TMI-93, Kyoto, Japan, 1993.

(Palmer & Hearst 94) D. Palmer and M. A. Hearst, *Adaptive sentence boundary disambiguation*, Report No. UCB/CSD 94/797.

(Papageorgiou et al. 94) H. Papageorgiou, L. Cranias and S. Piperidis, *Automatic alignment in parallel corpora*, Proc. of the 32nd Annual Meeting of the ACL, 1994.

(Reynar & Ratnaparkhi 97) J. C. Reynar and A. Ratnaparkhi, *A maximum entropy approach to identifying sentence boundaries*, Computational Linguistics Archive cmp-lg/9704002, 1997.

(Sadler & Vendelmans 90) V. Sadler and R. Vendelmans, *Pilot Implementation of a Bilingual Knowledge Bank*. Proc. of Coling, pp 449-451, 1990.

(Sato 92) S. Sato, *CTM: An Example-Based Translation Aid System*. Proc. of Coling, pp 1259-1263, 1992.

(Sato & Nagao 90) S. Sato and M. Nagao, *Toward Memory-based Translation*. Proc. of Coling, pp 247-252, 1990.

(Simard et al. 92) M. Simard, G. Foster and P. Isabelle, *Using cognates to align sentences in bilingual corpora*, Proc. of TMI, 1992.

(Sumita & Iida 91) E. Sumita and H. Iida, *Experiments and Prospects of Example-based Machine Translation*. Proc. of the 29th Annual Meeting of the Association for Computational Linguistics, pp 185-192, 1991.

(Sumita & Tsutsumi 88), E. Sumita and Y. Tsutsumi, *A Translation Aid System Using Flexible Text Retrieval Based on Syntax-Matching*. TRL Research Report, Tokyo Research Laboratory, IBM, 1988.