[*Translating and the Computer 21*. Proceedings… 10-11 November 1999 (London: Aslib)]

A Multi-level Framework for Memory-Based Translation Aid Tools

Stelios Piperidis, Christos Malavazos, loannis Triantafyllou
Institute for Language and Speech Processing
Language Technology Applications Department
Epidavrou & Artemidos 15125 Marousi, Greece
and
National Technical University of Athens
spip, christos, yiannis@ilsp.gr

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 machine translation (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.

In this paper, we will describe a multi-level 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.

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

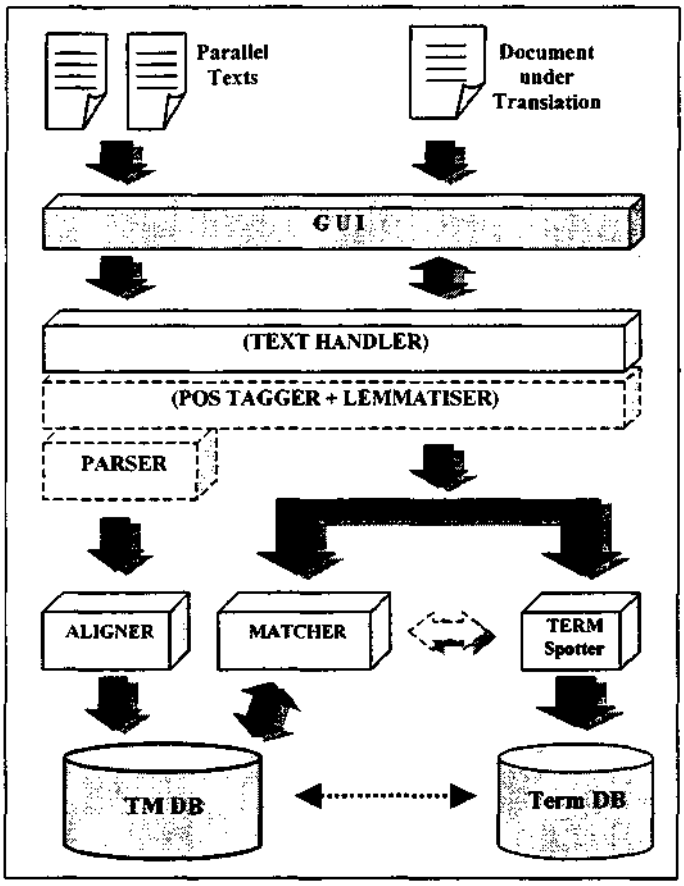

Figure 1: TrAID Architecture

2.1 Overview

Figure 1 displays TR•AID's architecture where all the individual components are presented within the overall framework. "Optional" components as well as optional links between different tools are presented in dotted lines. A detailed description of each individual system component will be provided in the following sections.

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.

Turning to translational equivalences below sentence level, the problems of low quality, as mentioned above, as well as ambiguity problems when the produced segments are rather short, become valid again. Despite the fact that most of the running EBMT systems employ the sentence as the text unit, it is believed that the potential of EBMT lies in 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). Along these lines, automatic sub-sentential alignment has started receiving attention lately (Boutsis & Piperidis 98). Their proposed 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, as described above, 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.

2.4 Underlying Database

The complexity inherent in the translation processes within a typical EBMT framework necessitates the existence of well-defined powerful resources. The need for optimal utilisation of different levels of available resources and the demand for real time responses, call for an efficient database architecture.

Approaches like the ones by Sato & Nagao (Sato & Nagao 90) or Watanabe (Watanabe 92), are characterised by rather complicated storage schemas (fully annotated structures) which consequently have a negative effect in the necessary storage and retrieval mechanisms in terms of computational cost as well as response time. (Sumita & Iida 95) proposed an alternative solution through parallel processing which was also adopted by other approaches.

Of critical importance, too, is the quality of the examples in the translation memory database, especially in terms of consistency. Conflicting translation examples should be identified and treated properly. Simple string matching techniques fail to overcome this problem. This matter has been addressed by (Nomiyama 92) and (Watanabe 94) under the term "exceptional examples". Furthermore, multiple occurrences of the same example should not add bias to the system.

In TR•AID, 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 phrase cross-language associations, should also be able to be accommodated later 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). Conflicting examples are identified and marked as such. The user can make the best choice through a ranked list proposed by the system. Precision can be improved if surface linguistic knowledge is added to the system (grammatical information).

A similar database schema has been adopted for managing terminological resources. Optimal utilization of different types of resources apart from efficient storage also necessitates uniform and seamless access to these resources as well as appropriate merging of results.

2.5 Text Matching

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 similar to the ones described in section 2.3.

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 quickly locates 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.

Two alternatives of the fuzzy match process have been examined bearing similar features but based on a different concept of similarity distance.

In the first approach, 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 second approach was based on 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.

Both approaches can be expanded to encapsulate surface linguistic information, in which case, the elements under comparison consist in a combination of word and lemma (and/or a pos tag) introducing, in this way, an intermediate level of similarity (instead of a binary one).

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 source language 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 input sentence which is again subsequently stored in the TM database. Thus, the translation memory system starts learning new translation pairs in an interactive mode.

Adaptation and reuse of the retrieved translation examples, is usually dealt with by integrating into the system conventional MT techniques (Kaji et al. 92), (Sumita & Iida 91). Partial modifications of the translation proposal, at the sub-sentence level (clauses/words or terms), should also be possible, provided that alignment of the translation archive at this level is available.

In a further attempt to "make the best" out of existing translation examples, many recent approaches induce explicit generalisations towards extraction of generalised translation templates (Kaji et al. 92), (Watanabe 92), (Furuse & Iida 92), (Furuse & Iida 96), (Somers et al. 94), (Carl et al. 98). These stand somewhere between example-based and rule-based approaches borrowing features from both. The definition of "translation units" in each particular method is based on different techniques varying from simple rule based processes to pattern matching or even psycholinguistic parameters (marker hypothesis) as well as combinations of the above. However, since "variable features" of text (grammatical information, subcategorisation information, etc) upon which generalization can be achieved, are only found within deeper linguistic information (as opposed to plain wordforms), such information is also required in order to correctly adopt the proposed target fragments. Matching at the syntax level, coupled by lexical similarity in a hybrid configuration, is believed to be the best an EBMT system could offer as a translation proposal.

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 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.

Term spotting is performed in two different ways depending on the available linguistic resources:

a) In case tagging and lemmatisation tools are available, input sentences are appropriately processed prior to term spotting. "Canonical forms" of the corresponding terms are searched in the database in a rather straightforward procedure.

b) However, considering the vast collection of languages and domains the system will be expected to deal with in real life applications the cost of acquiring or developing such tools may be prohibitive calling for more cost effective and practical solutions. To cater for this problem, an alternative approach known as "term conflation" has been proposed, during which the system aims at capturing morphological variations of terms located in the database (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. At this phase, the system extracts a small set of candidate terms based on statistical information. 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. Terms constitute basic translation units. In this respect, they should be treated as individual units during the matching and translation process. Prior to sentence matching terms are located and appropriately marked up. Sentence matching is performed as previously described but for the deviation that in this case terms enter the matching process as single tokens. In many cases this treatment seems to improve the precision of the matching process, however decreasing recall as well as response time due to the extra processing phase. Consequently, the benefits of this are questionable and dependent on the application at hand.

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. High-quality fully automatic machine translation is not yet feasible. Furthermore, it is generally believed that the future of MT (at least the near future) lies on the efficient merging of different MT engines towards the creation of hybrid multi-engine MT systems. To this respect, the goal should be to develop systems that optimally combine different levels of sophistication and resources and which will also be easily adaptable to different languages and domains.

References

(Boutsis & Piperidis 98) S. Boutsis, S. Piperidis, *Aligning Clauses in Parallel Texts,* 3rd Conference on Empirical Methods in Natural Language Processing, June 1998

(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.

(Carl et al. 98) M. Carl, L.L. lomdin, O. Streiter, *Towards a Dynamic Linkage of Example-Based and Rule-Based Machine Translation,* Proc. of the Machine Translation Workshop, 10th European Summer School in Logic, Language and Information, Saarbrücken, 1998.

(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.

(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.

(Furuse & Iida 96) O. Furuse and H. Iida, *Incremental Translation Utilizing Constituent Boundary Patterns.* Proc. COLING '96, pp 412-417, 1996.

(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.

(Nomiyama 92) H.Nomiyama, *Machine Translation by Case Generalization,* COLING 92, Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, Nantes, 714-720, 1992.

(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.

(Somers et al. 94) H. Somers, I. McLean, D. Jones, *Experiments in Multilingual Example-Based Generation,* CSNLP 1994: 3rd Conference on the Cognitive Science of Natural Language Processing, Dublin, 1994.

(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 & Iida 95) E. Sumita and H. Iida, *Heterogeneous Computing for Example-Based Translation of Spoken Language.* Proc. of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation, Leuven, Belgium, 273-286, 1995.

(Watanabe 92) H. Watanabe, *A Similarity Driven Transfer System.* COLING 92, Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, Nantes, 770-776, 1992.

(Watanabe 94) H. Watanabe, *A Method for distinguishing Exceptional and General Examples in Example-Based Transfer Systems,* COLING 94, The 15th International Conference on Computational Linguistics, Kyoto, 39-44, 1994.