Translation Memory vs. Example-based MT – What's the difference?

HAROLD SOMERS *(UMIST, Manchester)*

GABRIELA FERNANDEZ DIAZ *(Universidad de Sevilla)*

1 Introduction

Throughout the 1970s and 1980s, Machine Translation (MT) research focused on the development of so-called "second generation" systems, which aimed to translate text by a process of rule-driven linguistic processing, usually in three stages: syntactic-semantic analysis of the source text, bilingual transfer at a more or less abstract level of representation, and target-text generation from syntactic representation. At the same time, from a practical point of view, there was much discussion on how systems built with this architecture could be used to deliver a reasonable standard of translation usable by real users. Most popular were ideas of restricting the input (the sublanguage and controlled language approaches), or involving the user in pre- and post-editing. Interactive MT, where the user and the computer would cooperate in resolving ambiguities and making choices, was also championed.

In the early 1990s, with these ideas fairly well established, and perhaps even growing stale, research in MT was hit by an apparently new paradigm in which in particular the reliance on linguistic rule systems was to be (at least partially) replaced with the use of a corpus of already-translated examples which would serve as models to the MT system on which to base its new translation. This approach came to be known as Example-Based MT (EBMT) which had in fact first been proposed ten years earlier, in 1981. We will look in a little more detail at this development below.

At about the same time, a new tool for translators was being mentioned by developers. Like EBMT, it used a corpus of already-translated examples to serve as models for the new translation, but crucially, it was the human users, not the computer itself, who should determine exactly how to use the examples in producing a new translation. This tool is of course now widely known as a Translation Memory System (TMS).[1]

There are many commentators who regard EBMT and TMSs as essentially the same thing, and indeed certain developments in both have brought them closer to each other. The present author has long maintained that there is a crucial difference, which will be elaborated in the next sections. However, there are similarities, and in particular proposals to make TMSs better invariably make them look more and more like EBMT systems. This, principally, is the subject of this paper.

[1] We will throughout distinguish the system from its principal component, the database of stored translations, i.e. the "memory". For this reason we will refer to the systems as TMSs, and to the database itself as a TM.

2 Intertwined history of TMSs and EBMT

At a risk of repeating points that are familiar to readers of this journal, we wish in this section to describe some key moments in the development of the concept of TM(S)s. The original idea for a TM is usually attributed to Martin Kay's well-known "Proper Place" paper (1980), although the details are only hinted at obliquely:

> [T]he translator might start by issuing a command causing the system to display anything in the store that might be relevant to [the text to be translated] .... Before going on, he can examine past and future fragments of text that contain similar material. (Kay, 1980:19)

Interestingly, Kay was pessimistic about any of his ideas for what he called a "Translator's Amanuensis" ever actually being implemented. But Kay's observations are actually predated by the suggestion by Peter Arthern (1978)[2] that translators can benefit from on-line access to similar, already translated documents, and in a follow-up article, Arthern's proposals quite clearly describe what we now call TM(S)s:

> It must in fact be possible to produce a programme [sic] which would enable the word processor to 'remember' whether any part of a new text typed into it had already been translated, and to fetch this part, together with the translation which had already been translated, ...
>
> Any new text would be typed into a word processing station, and as it was being typed, the system would check this text against the earlier texts stored in its memory, together with its translation into all the other official languages [of the European Community]. ... One advantage over machine translation proper would be that all the passages so retrieved would be grammatically correct. In effect, we should be operating an electronic 'cut and stick' process which would, according to my calculations, save at least 15 per cent of the time which translators now employ in effectively producing translations. (Arthern, 1981:318).

Alan Melby (1995:225f) suggests that the idea might have originated with his group at Brigham Young University (BYU) in the 1970s. What is certain is that the idea was incorporated, in a very limited way, from about 1981 in ALPS, one of the first commercially available MT systems, developed by personnel from BYU. This tool was called "Repetitions Processing", and was limited to finding exact matches *modulo* alphanumeric strings.[3] The much more inventive name of "translation memory" does not seem to have come into use until much later.[4]

The first TMSs that were actually implemented, apart from the largely inflexible ALPS tool, appear to have been Sumita and Tsutsumi's (1988) ETOC ("Easy TO Consult"), and Sadler and Vendelman's (1990) Bilingual Knowledge Bank, predating work on corpus alignment which, according to Hutchins (1998) was the prerequisite for effective implementations of the TM idea. Also, Kugler et al. (1991) report on work by Keck (1989) based on statistical methods in the context of an ESPRIT research project. It is difficult to pinpoint when TMs entered the consciousness of translation studies researchers and translators in general. Brian Harris, introducing the notion of a "bi-text" in a translators' magazine, proposes something like a TM without using that

[2] Early proposals for a TM, and other aspects of the idea of a Translator's Workstation are described in Hutchins (1998).

[3] Curiously, this most innovative feature is barely mentioned in descriptions of the ALPS system, two exceptions being Sibley (1988:96,100) and Weaver (1988:121f) who mention the facility almost as an afterthought.

[4] Extensive enquiries have so far failed to produce a satisfactory identification of the first use of this term. Hutchins (1998:303) suggests that the Trados company were the first to use the term.

name (Harris, 1988:9):[5] a database of paired translations, searchable either by individual word, or by "a whole translation unit", in the latter case the search being allowed to retrieve similar rather than identical units. Cave (1988) responded to that article with an announcement that Logos were marketing just such a tool. In an unsigned 1991 article the magazine *Language International* reported that "text banks" had now made their appearance:

> The first of these would seem to have been IBM European Language Service's Translation Support Facility (TSF), which, …, incorporated a *repeated sentence identification facility*. (Anon, 1991:5; emphasis added).

The article goes on to explain the notion of "fuzzymatch" (sic) in the case where exact matches are not found.

The proceedings of Aslib's indicative annual conference series *Translating and the Computer* contain no mention at all of TMs until 1992, when three separate articles (Freibott, 1992; Le-Hong et al., 1992; Svanholm, 1992) mention them, in one case (Le-Hong et al.) without feeling the need to explain the term. Brace (1992) reported development of a TM tool by Trados, as well as the ESPRIT project mentioned above, and projects at IBM's European Language Services (Denmark) and the Official Languages and Translation sector of the Canadian Department of the Secretary of State in Ottawa.

The idea for EBMT has a similar chronology, with ideas surfacing in the early 1980s (the paper presented by Makoto Nagao at a 1981 conference was not published until three years later – Nagao, 1984), but the main developments being reported from about 1990 onwards.[6] The essence of EBMT, called "machine translation by example-guided inference, or machine translation by the analogy principle" by Nagao, is succinctly captured by his much quoted statement:

> Man does not translate a simple sentence by doing deep linguistic analysis, rather, Man does translation, first, by properly decomposing an input sentence into certain fragmental phrases ..., then by translating these phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference. (Nagao, 1984:178f)

Nagao correctly identified the three main components of EBMT: matching fragments against a database of real examples, identifying the corresponding translation fragments, and then recombining these to give the target text. Clearly EBMT involves two important and difficult steps beyond the matching task which it shares with TMS.

The idea of EBMT really took off in the early 1990s, with an increasing number of papers at conferences reporting on this approach. Pioneers were mainly in Japan, including Sato and Nagao (1990) and Sumita et al. (1990). Mention should also be made of the work of the DLT group in Utrecht, often ignored in discussions of EBMT, but dating from about the same time as (and probably without knowledge of) Nagao's work. The matching technique suggested by Nagao involves measuring the semantic proximity of the words, using a thesaurus. A similar idea is found in DLT's "Linguistic Knowledge Bank" of example phrases described in Pappegaaij et al.

[5] In the next paragraph, he describes it as providing "a memory-perfect exploitation of the translator's own previous experience".

[6] A thorough review of the literature on EBMT is attempted in Somers (1999).

(1986a,b) and Schubert (1986:137f). Sadler’s (1991) “Bilingual Knowledge Bank” clearly lies within the EBMT paradigm.

During this early period, individual researchers in the field used alternative names, perhaps wanting to bring out some key difference that distinguished their own approach: “case-based” (Collins and Cunningham, 1996), “analogy-based” (Nagao, 1984), and “experience-guided” (Zhao and Tsujii, 1999) are all terms that have been used. The first of these recalls approaches to Machine Learning known as “case-based reasoning” (Riesbeck and Schank, 1989), and other related models.[7] Another term found is “memory-based translation” (Sato and Nagao, 1990; Kitano, 1993), the use of which probably did most to suggest affinities between EBMT and TMSs.

3 What EBMT and TMSs could have in common

EBMT and TMSs have in common the use of a database of previous translations, the “memory” or “example-base”, and the essential first step, given a piece of text to translate, of finding in the example database the best match(es) for that text. Once the match has been found, the two techniques begin to diverge. However, it would be misleading to assume that all they have in common is the task of matching, or even that the approaches to matching in the two camps are particularly similar. Use of a database implies issues of database design, content, and maintenance. These will be the focus of the next sections.

3.1 How are examples found?

In TMSs, the TM database itself can be constructed in one of three ways. The simplest method, though the most time-consuming one, called “interactive translation” by Bowker (2002:108f) is to build a TM from scratch, that is, to store in the memory each sentence as it is translated. A second method, referred to by Bowker (2002:109f) as “post-translation alignment”, and much heralded by manufacturers, is to extract a TM from an already translated text by *aligning* the source and target texts. This can be a more or less irksome task (cf. Macdonald, 2001), and there is a considerable literature describing various alignment methods involving differing amounts of (linguistic) sophistication (see Manning and Schütze, 1999:466–486 or Wu, 2000a). O’Brien concludes:

> A translation memory is always more accurate when it has been created by interactive translation as opposed to automatic alignment, but alignment can produce a reasonably accurate translation memory which can be used as a start-up. (O’Brien, 1998:119)

Finally, TMs that have already been created can be imported, and the establishment of agreed interchange formats between manufacturers has hugely facilitated this (notably the Translation Memory eXchange (TMX) format developed by LISA (Localisation Industry Standards Association, cf. Melby, 1998, 2000 and Topping, 2000).

The size of the TM is an obvious question. The TM literature says little more than “the bigger the better”, subject to processing limitations, though Bowker (2002:108) warns that “size should not come at the expense of organization”, suggesting that separate TMs for different subject fields or clients may help to reduce false hits caused by homonymy. She adds:

[7] The relationship between EBMT and Case-Based reasoning is discussed in Somers and Collins (in press).

> Keep in mind that a larger TM will result in a greater number of matches .... Therefore, while it may seem logical at first glance to build a single large TM ..., this may turn out to be a false economy.... Moreover, there is a greater likelihood of retrieving "noise" (e.g., matches that are not helpful, matches containing homonyms) and the translator may waste a considerable amount of time analyzing, eliminating, or editing these poor matches. (*idem*.)

For Heyn (1998), a "big" TM will have between 100,000 and 1 million units, thanks to recent technology advances.

> In recent sparsely-coded-matrix based systems, real interactive work on 'big' master translation memories is possible. Big translation memories are typically in the order of 100,000 translation units, although memories in the range of 500,000 to 1,000,000 translation units are envisaged by the end of 1997. According to current research estimates, translation memories could be made up to 40% bigger without any increase in constant access times. (Heyn, 1998:128)

In the EBMT literature, the sizes of the example-bases reported vary over a huge range, with 0.73m the biggest, and 7 (seven!) the smallest reported (cf. Somers, 1999:120). Obviously, the systems with tiny example-bases are purely experimental, while the more serious systems will have thousands of examples (rather than, say, hundreds).

User manuals for TMSs suggest revising the database every so often to clear out useless examples. By "useless" is presumably meant "not used" rather than, for example, "misleading" (cf. Heyn, 1998:131f). It is easy to see how a TMS could incorporate a measure of the former, simply by counting access. To measure the latter, it would also need to "know" what the translator is doing with the proposed match. The "suitability" of examples is addressed in the context of EBMT systems by various researchers. Nomiyama (1992) introduces the notion of "exceptional examples", an idea further developed by Watanabe (1994). As far as can be seen, these examples are exceptional just in the sense that if used they give the wrong result! Clearly a more systematic notion is needed. It is well known that the same phrase can be translated differently in different circumstances. Ellipsis, anaphora and stylistic variation can contribute to this, in which case different examples may be seen as nonetheless equivalent in some sense. On the other hand, the underlying meaning of a phrase may differ depending on the context. Somers et al. (1990:274) illustrate how the simple phrase *OK* in a conversation may be translated into Japanese as *wakarimashita* 'I understand', *iidesu yo* 'I agree' or *ijō desu* 'let's change the subject'.

There is also an issue of "granularity": both in TM and EBMT, there is a trade-off between length and similarity of examples. The longer the example units, the lower the chance of an exact match; but the shorter the units, the greater the probability of "ambiguity" (multiple, conflicting, matches), with a corresponding decrease in the quality of the proposed translation. Nirenburg et al. (1993:48) call this "passage boundary friction and incorrect chunking". The obvious and intuitive "grain size" for examples, to judge from almost all TMSs and EBMT systems, is the sentence, though evidence from translation studies (Gerloff, 1987; McTait et al., 1999) suggests otherwise: human translators process text in "naturally-occurring syntactic units" and "generally there is very little processing at sentence level" (McTait et al., 1999). According to Bennett,

> ... there are good reasons for keeping the U[nit of] T[ranslation] (in the sense of translation atom) in MT as small – and hence as manageable – as possible. Adopting a larger UT may be less efficient, and is not guaranteed to improve translation quality. (Bennett, 1994:18)

the "translation atom" being the smallest segment that must be translated as a whole (*ibid*., p.13). Schäler et al. (2003) echo this sentiment, suggesting that "matching segments at sentence level unnecessarily restricts the potential and the usefulness of translation memories" (p. 89), and propose "phrasal matching" as the primary mechanism for TMSs. Simard (2003) studied how real users make use of a bilingual concordancer, a tool which closely resembles a TMS in function, except that users can look up arbitrary sequences of words. He found that most users look up syntactically well-formed "chunks", and implemented a system based on this principle. In fact, matching segments rather than whole sentences produces far too many "hits", so the system must also have a way of selecting the most useful from amongst them. An evaluation of Simard's implementation suggested that it proposed between 15 and 30 times more "reusable material" than a sentence-based system.

Both EBMT and TMSs could probably be improved by concentrating on a more flexible view of the unit of matching/translation, and "the exploitation of fragments of text smaller than sentences" (Cranias et al., 1994:100). In TMSs, this idea is partly addressed in that terminology look-up is often seen as an integral part of the tool, although terminology tools are generally implemented in a lexicon-based rather than a memory-based manner. We will return to the issue of "fragments" below. According to Bowker,

> Many TM systems allow the user to define other units of segmentation in addition to sentences. These units can include sentence fragments or even entire paragraphs. (Bowker, 2002:94)

Along the same lines, Esselink states:

> A segment is a text element, which is considered by the application as the smallest translatable unit, defined by periods, semi-colons, and hard returns. These are usually sentences, but can also be chapter headings or items in a list. ....Translation memory tools usually allow the user to change and customize segmentation rules. (Esselink, 2000:362f)

3.2 How are examples stored?

In TMSs, examples are generally stored as plain text, sometimes with formatting information. Systems differ as to how they treat formatting (i.e. fonts, capitalization and so on) even though it is potentially very useful for matching (see below). Austermühl comments that

> Some translation memories have a built-in interface that works with common word-processors ... the format in which the translated text is stored in the translation memory is identical to that used in the word-processing program. (Austermühl, 2002:138)

From this we must infer that segments always keep their original format when stored.

A different issue is the way TMS deal with tags. Nowadays, translation tools are being used extensively in the software localization industry. For this reason, the new generation of TMSs such as Trados, Transit and Déjà Vu contain a wide range of filters to convert files from one format to another. At the same time, TMSs are designed to handle a wide variety of formats such as HTML, SGML and XML. Esselink confirmed the following in the year 2000:

> Most translation memory tools have standard filters for HTML files. HTML files usually contain very repetitive text, so it is worthwhile using translation memory, because of the substantial time and cost savings. Furthermore, translating updates of web sites is much easier and quicker if a translation memory of the previous version exists.
>
> Examples of translation memory tools that support the HTML format are Trados Translator's Workbench, IBM TranslationManager, STAR Transit, SDLX, and Atril Déjà Vu. (Esselink, 2000:218)

In 2003 new versions of TMSs have spread throughout the market. These new versions are well equipped to deal with every kind of application for the design of web pages, presentations, graphics, etc. Thus, Transit XV includes a number of filters that make it possible to translate files generated with programs such as Excel, PowerPoint, QuarkXPress, PageMaker, FrontMaker and AutoCAD, among others. In the same way, the latest solutions presented in the market by Trados and Déjà Vu, namely, Trados 6.5 and Déjà Vu X, incorporate filters which allow the user to import and export files with any kind of format.

Increasingly, TMS developers are recognising the value of incorporating "mark-up" into their systems, not just formatting but also linguistic annotations such as syntactic part-of-speech (POS) tags. In this respect, Planas (1999:8), states that the Xerox XMS Memory Manager, is a "linguistically based tool", and consequently is capable of retrieving better matches than character-based systems. We read "this shows the crucial importance of using linguistic data for enabling more precise retrieval of the closest sentence in the database". This system is currently still in the experimental stage however. Planas and Furuse (1999) proposed a much more elaborate scheme in which examples are represented in a multi-level lattice, combining typographic, orthographical, lexical, syntactic and other information. A major drawback of the most successful TMSs available in the market has to do with the lack of incorporating linguistic knowledge in their products.

Obviously, storage and matching methods are intricately related: we will return to the latter in the next section.

In EBMT systems, a wide range of formats have been proposed for storing the examples. Given its origins as a variant of rule-based MT, early EBMT systems supposed that examples would be stored as aligned tree structures such as the one illustrated in Figure 1, from Watanabe (1992).

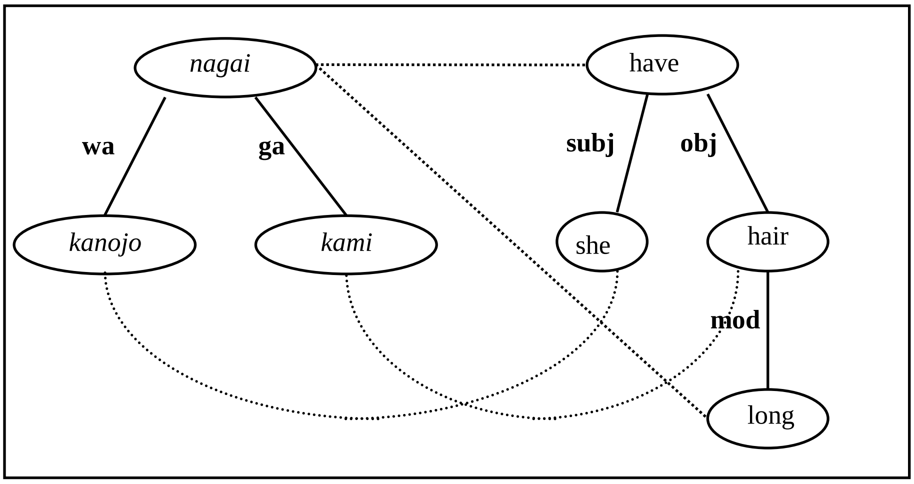


Figure 1. Representation schema for Kanojo wa kami ga nagai *(lit. 'she* TOPIC *hair* SUBJ *is-long')* ↔ She has long hair.

Unfortunately, such a representation involves serious overheads in storage space, analysis at run time, and verification of structures, a criticism that also applies to Planas and Furuse's proposal, as shown in Figure 2. The resulting reliance on parsing or other knowledge-rich processes is acknowledged as a disadvantage.

Because these rich representations are widespread in early EBMT proposals, they are often thought of as being a *necessary* feature of EBMT, though this is quite incorrect. Later EBMT proposals involve much less ambitious representation schemas, in particular lightly annotated text in which words are accompanied by POS tags and/or the results of "stemming" (i.e. morphological analysis to identify root or stem, and partially interpret endings). Planas (1999:8) illustrates the idea by considering sentence (1a) compared to each of (1b-d): although (1c) differs by only one character, humans instinctively find (1b) a better match.

(1) a. The white horse is nice.
b. The white horses are nice.
c. The white house is nice.
d. The white houses are nice.

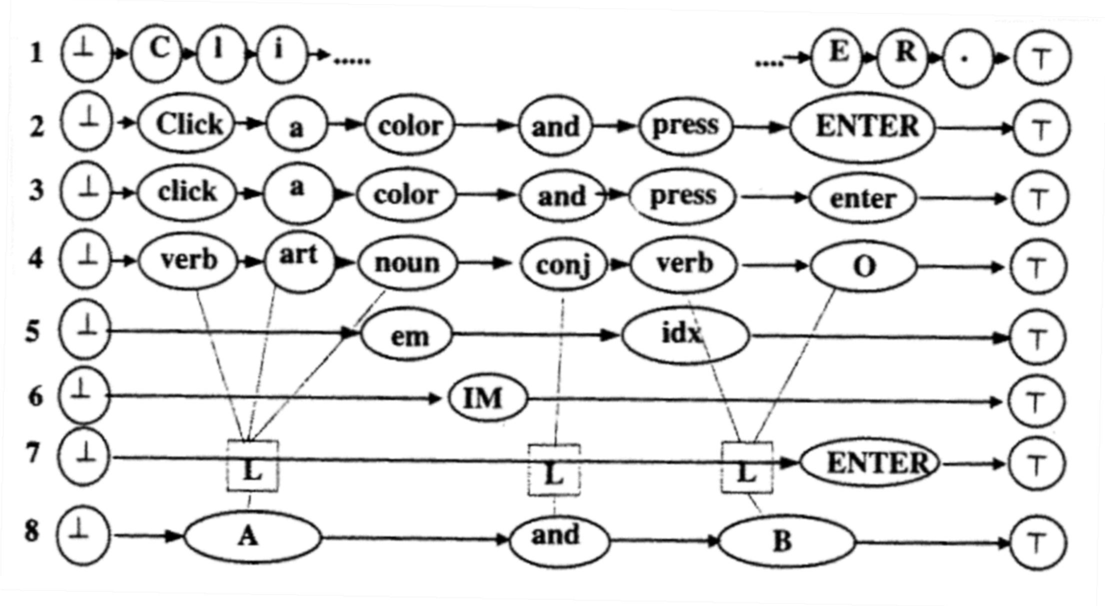


Figure 2. Structured representation suggested for 'Click a color and press ENTER.' *(from Planas and Furuse, 1999, p. 333).*

An important recent trend in EBMT is to store similar examples in a unified "generalized" manner. For instance, the two examples in (2a,b) could be generalized as (2c), and stored as such, with obvious repercussions for matching (see next section).

(2) a. John Miller flew to Frankfurt on December 3rd.
b. Dr Howard Johnson flew to Ithaca on 7 April 1997.
c. <person-m> flew to <city> on <date> .

An early proposal along these lines is found in Furuse and Iida's (1992) distinction between "literal examples" and "pattern examples", the latter containing variables in place of words, with the variables characterized by a (list of) typical filler(s), as in (3).[8]

[8] They also have a third type, called "grammar examples" which consist almost entirely of variables, rather like the rules in conventional MT.

(3) a. *X o onegai shimasu* → may I speak to the X′ (X=*jimukyoku* ‘office’, …)
b. *X o onegai shimasu* → please give me the X′ (X=*bangō* ‘number’, …)

The idea is quite widespread in the EBMT literature, including Kaji et al.’s (1992) “pseudo-sentences”, Langé et al.’s (1997) “skeleton sentences” and a number of others.[9] The examples are usually generalized by merging similar cases, though authors differ as to whether this can be done (semi-)automatically, or manually. Certainly, if the idea was applied to TMSs, TMs could be much reduced in size, though access would presumably have to be more sophisticated.

3.3 Matching techniques

Accessing the TM or example-base involves “matching” the given translation unit against the cases already stored. Early implementations of TMSs could handle only exact matches, although alphanumeric “replaceables” as in (4) were allowed.

(4) a. This is shown as A in the diagram.
b. This is shown as B in the diagram

Bowker introduces the distinction between an “exact” match and a “full” match.

> An exact match is 100 percent identical to the segment that the translator is currently translating, both linguistically and in terms of formatting. ... This means that the two strings must be identical in every way, including spelling, punctuation, inflection, numbers, and even formatting (e.g., italics, bold). (Bowker, 2002:96f)

Thus (5b) would not be retrieved as an exact match for (5a) because of the difference in formatting.

(5) a. Click on OK
b. Click on *OK*.

A “full match” on the other hand,

> … occurs when a new source segment differs from a stored TM unit only in terms of so-called variable elements, which are sometimes referred to as “placeables” or “named entities”. Variable elements include numbers, dates, times, currencies, measurements, and sometimes proper names. (*ibid*., p. 98)

Bowker’s “placeables” have been termed “transwords” by Gaussier et al. (1992), while Macklovitch and Russell (2000) call them “non-translatables”. The notion of “named entities” is found in Information Retrieval. As the latter authors point out, they are treated in translation in a rather transparent manner, either not translated at all, or subject to specific conventions, and in any case, independent of context. For a TMS their impact is twofold. On a simple level, we want to have matchers that “ignore” them, so that (4a,b) above are effectively “exact matches”. Additionally, in a more sophisticated TMS (and in EBMT), they are important building blocks for suggesting automatically a likely translation of the given text. Consider (6a) as a text to be translated, which matches with (6b), with the differences highlighted, and its associated translation (6c). In (6b) there are two transwords the “translation” of which can be readily identified in (6c), but you have to know the target language to know which word to change in (6c) to accommodate the lexical difference *large* vs. *small*.

[9] Nomiyama (1992), Almuallim et al. (1994), Akiba et al. (1995), Collins and Cunningham (1995), Jain et al. (1995), Matsumoto and Kitamura (1995), Watanabe and Takeda (1998), Carl (1999) – see Somers (1999:139ff).

(6) a. The large paper tray holds up to 400 sheets of A3 paper.
b. The small paper tray holds up to 300 sheets of A4 paper.
c. *Die kleine Papierkassette fasst bis zu 300 Blatt in A4-Format*

Some TMSs claim to use sophisticated matching algorithms (e.g. Trados's claimed use of "neural networks") and Heyn states that

> Modern computer science does, however, offer some workable solutions to similarity problems using fuzzy processing. These approaches include the use of neural networks and sparsely coded matrices. Whereas the first generation of the Trados translation memory system, for example, was based on a classical binary approach, and (linguistically motivated) substring operations on classical database indices, the current generation employs sparsely coded matrices. The advantages are obvious: phenomena like misspellings and complicated syntactic deviations are now manageable and access time has been reduced significantly. (Heyn, 1998:127)

However, all the evidence so far is that matching in TMSs is essentially a straightforward implementation of a character-based "edit distance", that is, the widely used measure of string similarity which counts the minimum number of substitutions, insertions and deletions needed to change one string into another.[10] Example (1) above illustrates the problem with this approach, as does (7), where (7b) differs from (7a) by only four letters, compared to nine for (7c).[11]

(7) a. The wild child is destroying his new toy.
b. The wild chief is destroying his new tool.
c. The wild children are destroying their new toy.

Consider also examples like the sentences in (8): as a match for (8a), the edit distance algorithm will prefer (8c) over (8b) because of the additional words. Similarly, (8b) and (8d) should be considered more similar than (8a) and (8c), because they contain more text in common; but the simple edit distance measures only differences, not similarities.[12]

(8) a. Select 'Symbol' in the Insert menu.
b. Select 'Symbol' in the Insert menu to enter a character from the symbol set.
c. Select 'Paste' in the Edit menu.
d. Select 'Paste' in the Edit menu to enter some text from the clipboard.

Proposals for a search mechanism for TMSs that takes syntactic and/or semantic information into account have been made by various authors. Dennett (1995) makes two proposals for TMSs: the first takes into consideration the relative significance of the words that have been changed, possibly on the basis of statistical data; the second identifies syntactically significant portions of the segment. Cranias et al. (1997) propose a scheme for both TMSs and EBMT that makes special use of function words as well as POS tags, and looking at lemmas rather than strings. Planas and Furuse (1999) suggest a flexible multi-layer matching scheme as indicated by their structure (seen in Figure 2, above). Macklovitch and Russell (2000) and Rapp (2002) similarly suggest taking inflection and syntactic category into account.

[10] So-called "fuzzy match" scores may take into account other superficial differences such as formatting, or the source of the example, but it is unlikely that any *linguistic* sophistication is involved, despite the manufacturers' claims.

[11] These examples are from Planas and Furuse (1999:331).

[12] These examples are from Somers (1999:129).

Interestingly, in EBMT it was always assumed that matching would be on a more sophisticated basis. The earliest proposals (e.g. Nagao 1984; Sumita et al. 1990; Sumita and Iida 1991) involved a thesaurus to measure semantic proximity of structures, as illustrated by Nagao's original example, where (9a,b) might be stored as examples (along with their Japanese translations as shown – the key being the different choice of *taberu* vs. *okasu* for 'eat'). If we wish to translate (9c), then (9a) is chosen as the model because of the better match between *man* and *he*, and *vegetables* and *potatoes*. Conversely, (9b) is a better model for translating (9d).

(9) a. A man eats vegetables. *Hito wa yasai o taberu*.
b. Acid eats metal. *San wa kinzoku o okasu*.
c. He eats potatoes.
d. Sulphuric acid eats iron.

Other EBMT developers (like Cranias et al. 1997, already mentioned) have proposed using POS tags or treating function words in a particular way (Furuse and Iida 1994; Veale and Way 1997). In the multi-engine Pangloss system, the matching process successively "relaxes" its requirements, until a match is found (Nirenburg et al. 1993): the process begins by looking for exact matches, then allows some deletions or insertions, then word-order differences, then morphological variants, and finally POS-tag differences, each relaxation incurring an increasing penalty.

Chatterjee (2001) proposes an evaluation scheme where a number of different features, differentially weighted, combine to give a score which reflects similarity at various levels: lexical, morphological, syntactic, semantic, pragmatic. The strength of EBMT, especially for dissimilar language pairs, is in using examples with a similar meaning, rather than a similar structure, so that the semantic and pragmatic features, which can still be captured by simple morphosyntactic features (e.g. whether the subject of the verb is animate) are weighted heavily.

Earlier proposals for EBMT, and proposals where EBMT is integrated within a more traditional approach to MT, assumed that the examples would be stored as structured objects, so the process involves a rather more complex tree-matching (e.g. Maruyama and Watanabe 1992; Matsumoto et al., 1993) though there is generally not much discussion of how to do this (cf. Maruyama and Watanabe 1992; Al-Adhaileh and Tang 1998), and there is certainly a considerable computational cost involved, so this is probably a step too far for TMSs.

3.4 How many matches?

The final issue to be discussed in this section is the question of how the matches are presented to the user. The standard method in TMSs is to present the single best match, with an indication of its "score" and the possibility of other matches being available with a lesser score, or else to show, in a scrolling window, a range of matches, ordered once again by score. Bowker (2002:102) gives a nice example, partly reproduced here, in which the source segment (10a) matches best against (10b), then (10c) and so on. The exactly matching parts of the retrieved examples are highlighted.

(10) a. The operation was interrupted because the file was hidden.
b. The operation was interrupted because the Ctrl-c key was pressed.
c. The specified method failed because the file is hidden.
d. The operation was interrupted by the application.
e. The requested operation cannot be completed because the disk is full.

The examples in (10) show very nicely a feature that would be very desirable in TMSs, but so far has only been a promise, with one notable exception. The translation of (10a) could actually be achieved by selecting the appropriate parts of two examples retrieved from the TM: the first part of (10b) linked to the second part of (10c). This is the idea of "sub-segment matches", which, as Bowker (2002:103) points out, "falls partway between fuzzy and term matching". The main difference is that the segments identified in this manner would not have the status of terms; indeed, the sub-segments would be identified on a case-by-case basis, and do not even have to correspond to complete phrases, as illustrated in (10b), where the sub-segment ends in the middle of the noun phrase.

Simard and Langlais (2001) explicitly suggest that this idea be incorporated into TMSs, referring to the EBMT literature where it has been around for a while. Simard (2003) explains the scheme in more detail. Nirenburg et al. (1993) referred to "substring" matching, Somers et al. (1994) preferred the term "fragments", while Collins (1998) talked of "chunks". Figure 3 illustrates the idea.

```
Source sentence:
there is a danger of avalanche above 2000m
Matching fragments:
danger of N < > above, danger of, of N < > above,
above CRD m, there is a, avalanche < > above,
there is, is a, danger of avalanche,
avalanche above CRD m, avalanche above,
of avalanche, there is < > a,
is < > a, there is a < > danger < > of,
there is < > danger < > of, there is a < > danger,
a < > danger,  there is < > danger
```

Figure 3. Fragments extracted for the input there is a danger of avalanche above 2000m. *The individual words are POS-tagged (not shown here); the matcher can also match tags only, and can skip unmatched words, shown as* < >. *N is noun, CRD is cardinal number. Adapted from Somers et al. (1994).*

Simard and Langlais suggest concentrating on "linguistically motivated" sub-sequences, though this implies that the matcher must include some "linguistic knowledge". Merkel et al. (1994) describe a system which extracts "recurrent segments" from a corpus, storing only the longest possible segments where there is overlap.

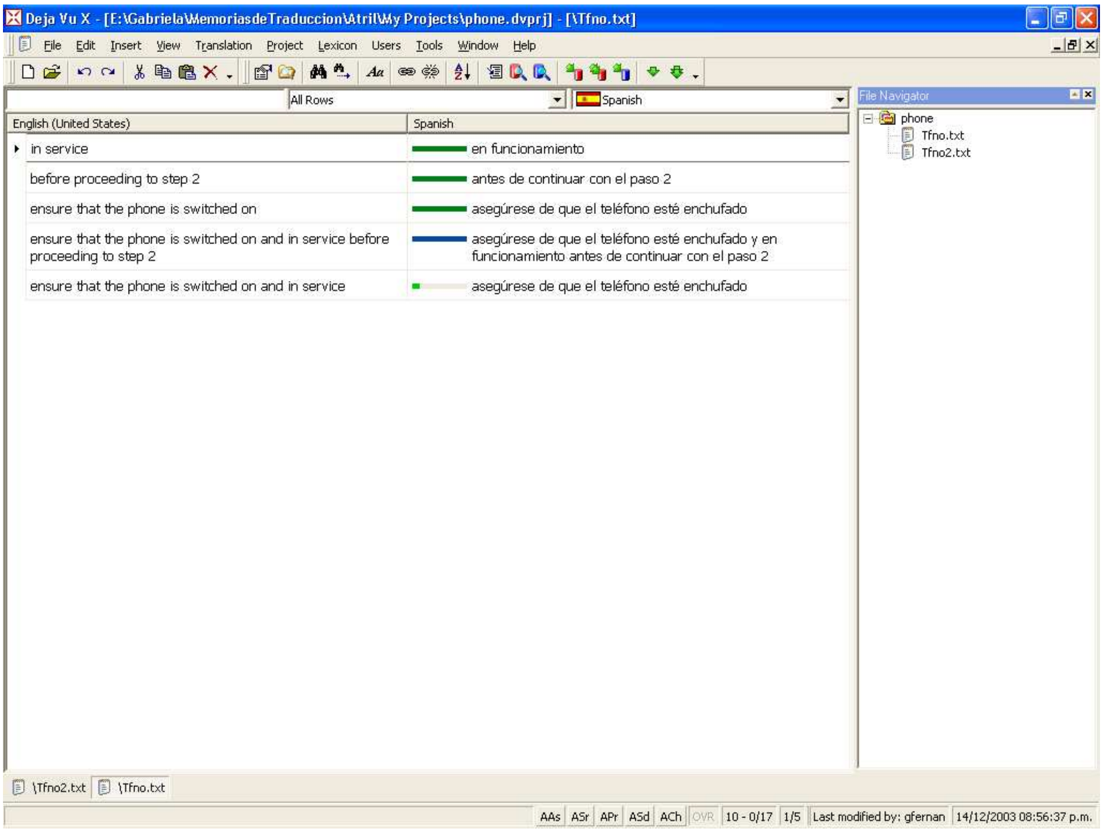
One commercial TMS does offer the possibility of assembling a translation from fragments: Déjà Vu, in its latest version, Déjà Vu X, talks about EBMT technology when describing the functionality of their system. In the user's manual they claim:

> We use the term *example-based machine translation* to describe Déjà Vu X's unique ability to self-repair fuzzy matches from the translation memory by deleting the incorrect part of the sentence and replacing it with the correct one. Provided that Déjà Vu X has sufficient terminology databases, it is able to do this through the close association of the memory matching and assemble processes. [emphasis original]

Déjà Vu names this facility "assemble from portions". The facility makes it possible to convert fuzzy matches into exact matches automatically.

Figure 4 shows the system assembling the translation (11) (marked with a blue line in the figure) from the three portions indicated (marked with green lines in the figure).

(11) Ensure that the phone is switched on and in service before proceeding to step 2.
Asegúrese de que el teléfono esté encendido y en funcionamiento antes de continuar con el paso 2.


Deja Vu X - [E:\Gabriela\MemoriasdeTraduccion\Atril\My Projects\phone.dvprj] - [\Tfno.txt]
File Edit Insert View Translation Project Lexicon Users Tools Window Help
All Rows
Spanish
File Navigator
phone
Tfno.txt
Tfno2.txt
\Tfno2.txt
\Tfno.txt
AAs ASr APr ASd ACh OVR 10 - 0/17 1/5

English (United States)	Spanish
in service	en funcionamiento
before proceeding to step 2	antes de continuar con el paso 2
ensure that the phone is switched on	asegúrese de que el teléfono esté enchufado
ensure that the phone is switched on and in service before proceeding to step 2	asegúrese de que el teléfono esté enchufado y en funcionamiento antes de continuar con el paso 2
ensure that the phone is switched on and in service	asegúrese de que el teléfono esté enchufado

Last modified by: gfernan 14/12/2003 08:56:37 p.m.

Figure 4. Déjà Vu's "Assemble from portions" facility.

However, it must be reported that our experience[13] the conditions have to be very favourable for this facility to work. For example, the portions have to be roughly equal in length and "importance": we were unable to get it to offer a translation from portions of (12) (marked with an incomplete green line in the figure), where the first portion represents too high a percentage of the total sentence. Also, it appears that all the portions have to be in the TM, and have to match exactly (even *and* = *y* must be in the TM) before it will work satisfactorily.

(12) Ensure that the phone is switched on and in service.

We should not be too harsh on Déjà Vu, because actually subsegment matching and reassembly represents the most difficult technical problem for EBMT. Notice once again the difference between TMSs and EBMT: in a TMS it is up to the user to decide what to do with the matches, whereas in EBMT the system must operate automatically. Consider again (10b,c), this time with their associated translations, reproduced here as (13).

(13) a. The operation was interrupted because the Ctrl-c key was pressed.
L'opération a été interrompue car la touché Ctrl-c a été enfoncée.
b. The specified method failed because the file is hidden.

[13] We are grateful to Luis Cerezo Ceballos for his work with Déjà Vu.

La méthode spécifiée a échoué car le fichier est masqué.

The user (or system) of course has to know French to be sure which parts of the translations correspond to the underlined segments, *and* to know how to fit them together. In EBMT this is known as “recombination”, and involves the “boundary friction” problem. Assuming one can correctly identify the target-language fragments associated with the source-language fragments (and this is not necessarily trivial), it may not be the case that they can be simply “glued” together. This is better illustrated with a language like German. Suppose for example we extract the translation associated with *the handsome boy* from (14a): it is not equally reusable in either of the sentences in (14b,c), since in German the form of the determiner, adjective and noun can all carry inflections to indicate grammatical case (14d,e).

(14) a. The handsome boy entered the room.
Der schöne Junge kam ins Zimmer.
b. The handsome boy ate his breakfast.
c. I saw the handsome boy.
d. *Der schöne Junge ass seinen Frühstück.*
e. *Ich sah den schönen Jungen.*

4 Where EBMT and TMSs differ

So far we have looked at a range of issues that are more or less common to TMSs and EBMT. There are important differences however, mainly stemming from the fact that a TMS is a translator’s aid, where the user has the main responsibility for making decisions, whereas EBMT is a way of doing translation automatically.

The main difference then lies in the fact that a TMS has essentially just the single step of matching examples, while EBMT must then *do so*mething with the matches found. As Ahrenberg and Merkel (1996) state:

> The ddifference between a translation memory as a support tool for the translator and a full-feldged example-based system is thus basically a difference in who has the prime responsibility for drawing analogies and structur[ing] the target text during translation.

What EBMT does consists of two steps, often referred to as alignment and recombination.

4.1 Automatic alignment of matches

We have already referred to the *alignment* problem, although not by name as such.[14] This is the task of knowing which parts of the (partial) match are relevant for the translation. In particular, examples (6) and (13) were used to illustrate the point.

In most EBMT systems, the solution lies in the way the examples are stored. In early systems, as we have already noted, examples are stored as linked structured representations, as exemplified in Figure 1. Another approach, seen for example in Somers et al. (1994), is to extract common elements from multiple matches. Looking again at Figure 2, it might be that the phrase *there is a danger* occurs in numbers of slightly different examples, but we can try to extract the target-language equivalent of this phrase because it will (presumably) recur in the translations paired with the retrieved examples. This is one of the basic ideas used in efforts to extract bilingual

[14] Not to be confused with the process, also called “alignment”, of converting a parallel text into a TM. There are of course some common aspects to these two similarly named processes.

lexical alignments from parallel texts (cf. Véronis 2000). The techniques often involve statistical measures of co-occurrence, though reliance on such measures alone is not generally enough, and many authors try also to incorporate some linguistic information into the process, making use of a bilingual dictionary (e.g. Kaji et al. 1992; Matsumoto et al. 1993) or existing MT lexicon, as in the cases where EBMT has been incorporated into an existing rule-based architecture (e.g. Sumita et al. 1990; Frederking et al. 1994).

EBMT however goes beyond this kind of lexical alignment, since the examples retrieved are sources not just of lexical equivalence information, but also serve as *models* for the structure of the target text.

4.2 Recombination

This is where the second step comes in: the target-language fragments suggested by the examples then have to be reassembled or *recombined* to form the target text. Where EBMT is combined with more traditional methods, the system might include a target-language grammar which could iron out any difficulties in creating the target text. More recently, such grammars have been derived automatically from parallel corpora, as is the case with Stochastic Inversion Transduction Grammars (Wu 2000b). Simard and Langlais (2001) report experiments with this formalism. Other researchers discuss the problem and in general seem to agree that some form of grammar formalism is needed (e.g. Carl 2001; Way 2001).

4.3 Generalized examples

A major trend in EBMT is to try to form general translation (or “transfer”) rules from the examples. We have already seen this illustrated in (2) above, where the generalized examples are presumably constructed manually, but a number of researchers have proposed doing this automatically on the basis of “minimal pairs”, i.e. pairs of sentences that contrast in a minimal way, and from which some generalization can be inferred. Consider the English–Turkish sentence pairs in (15) (from Cicekli and Güvenir 1996; Güvenir and Cicekli 1998) or the English–Spanish pairs in (16) (from McTait et al. 1999; McTait 2001).

(15) a. I took a ticket from Mary ↔ *Mary’den bir bilet aldım*
b. I took a pen from Mary ↔ *Mary’den bir kalem aldım*

(16) a. The Commission gave the plan up ↔ *La Comisión abandonó el plan*
b. Our Government gave all laws up ↔ *Nuestro Govierno abandonó todas las leyes*

From the sentence pairs can be identified the common elements, which are supposed to be mutual translations (17). This generalization can be stored as a translation “template”.

(17) a. I took a … from Mary ↔ *Mary’den bir … aldım*
b. … gave … up ↔ *abandonó*

The complementary elements in the matched sentences can be supposed to correspond as shown in (18).

(18) a. ticket ↔ *bilet*; pen ↔ *kalem*
b. The Commission … the plan ↔ *La Comisión … el plan*
Our Government … all laws ↔ *Nuestro Govierno … todas las leyes*

In the case of (18b), there is more work to be done, since (notwithstanding knowledge of Spanish or recognition of cognates), we have to establish how the remaining elements match up: this can be done by looking at further examples which isolate the words in question.

Of course much of this work could be simplified with the help of an on-line dictionary, assuming we had one. But what is of interest to researchers is the extent to which it can be automated. To exemplify this, consider the examples in (19), in a language probably unfamiliar to most readers.

(19) a. *Dia nak pěrgi kě kědai běli roti*.
She is going to go to the shops to buy bread.
b. *Dia pěrgi kě pasar nak běli baju.*
She went to the market to buy a shirt.
c. *Měreka pěrgi kě kampung nak běli kereta.*
They went to the village to buy a car.

It is not difficult to identify the probable word-pairings and from the examples to construct the correct translations of sentences like those in (20), and we invite the reader to try it as an exercise.[15] In doing so it should be noted how much generic (common-sense) knowledge about how languages work we as humans bring to this task, which may have to be simulated in an otherwise purely automatic system.

(20) a. She went to the village to buy bread.
b. They are going to the market.

5 TMSs would be better if they were more like EBMT

Our purpose in this paper has been to point out how some of the ideas developed in connection with EBMT could be introduced into the development of TMSs. In this final section we attempt to summarize the main proposals.

- If they could identify what in the target part of the match has to be changed.

> If we want to be able to match similar sentences, such as those differing only in the form of their words, we need linguistic analysis. We do not need a deep analysis that would take a long time to process, but just a “stemming” and a “tagging” one that would give a light but crucial analysis (Planas 1999:9).

> [T]he most promising strategy for the next generation of TM systems will be to employ various partial parsing or “chunking” techniques. (Macklovitch, 2000)

- If they could make suggestions about what in the target part of the match has to be changed to.
- If they could construct target texts from matched fragments.
- If they could take similar examples and make generalizations about them.

> TM and EBMT can be seen to lie at opposite ends of a spectrum in memory-based translation. On the one hand, TM requires few linguistic[ ] resources but cannot combine fragments from different T[ranslation] U[nit]s, and on the other hand, EBMT can combine example fragments, but does so by relying on … knowledge-intensive tools. (McTait et al. 1999)

[15] The language is Malay. The correct translations are *Dia pěrgi kě kampung nak běli roti*; *Měreka nak pěrgi kě pasar*.

References

Ahrenberg, L. and M. Merkel: 1996. ‘On Translation Corpora and Translation Support Tools: A Project Report’, in K. Aijmer, B. Altenberg, and M. Johansson (eds), *Languages in Contrast, Papers from a Symposium on Text-based Cross-linguistic Studies*, Lund: Lund University Press, pp. 185–200.

Akiba, Y., M. Ishii, H. Almuallim and S. Kaneda: 1995. ‘Learning English Verb Selection Rules from Hand-made Rules and Translation Examples’, in *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium, pp. 206–220.

Al-Adhaileh, M. H. and Tang E. K.: 1998. ‘A Flexible Example-Based Parser Based on the SSTC’, in *COLING-ACL ‘98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Quebec, pp. 687–693.

Almuallim, H., Y. Akiba, T. Yamazaki, A. Yokoo and S. Kaneda: 1994. ‘Two Methods for Learning ALT-J/E Translation Rules from Examples and a Semantic Hierarchy’, in *COLING 94, The 15th International Conference on Computational* Linguistics, Kyoto, Japan, pp. 57–63.

Austermühl, F. 2001. *Electronic Tools for Translators*. Manchester: St. Jerome.

Anon: 1991. ‘Bitext makes progress with and without the name’, *Language International* **3**.2:5–6.

Arthern, P. J.: 1978. ‘Machine Translation and Computerized Terminology Systems: A Translator’s Viewpoint’, in B.M. Snell (ed.) *Translating and the Computer: Proceedings of a Seminar, London, 14th November 1978*, Amsterdam (1979): North Holland, pp. 77–108.

Arthern, P. J.: 1981. ‘Aids Unlimited: The Scope for Machine Aids in a Large Organization’, *Aslib Proceedings*, **33**:309–319.

Bennett, P. 1994. ‘The Translation Unit in Human and Machine’, *Babel* **40**:12–20.

Bowker, L. 2002. *Computer-Aided Translation Technology. A Practical Introduction*. Ottawa: University of Ottawa Press.

Brace, C.: 1992. ‘The Many Flavors of Translation Memory’, *Language Industry Monitor* (May/June 1992); available at http://www.lim.nl/monitor/memory.html.

Carl, M.: 1999. ‘Inducing Translation Templates for Example-Based Machine Translation’, in *Machine Translation Summit VII*, Singapore, pp. 250–258.

Carl, M.: 2001. ‘Inducing Translation Grammars from Bracketed Alignments’, in *MT Summit VIII Workshop on Example-Based Machine Translation*, Santiago de Compostela, Spain, pp. 12–23; repr. in Carl and Way (2003), pp. 339–361.

Carl, M. and A. Way (eds): 2003. *Example-Based Machine Translation*, Dordrecht: Kluwer.

Cave, J.R.: 1988. ‘Observations on Bi-Text’, Letter to the Editor, *Language Monthly* **57**:18–19.

Chatterjee, N.: 2001. ‘A Statistical Approach for Similarity Measurement Between Sentences for EBMT’, in *STRANS–2001 Symposium on Trabnslation Support Systems*, Kanpur, India. [no page numbers]

Cicekli, I., and H. A. Güvenir: 1996. ‘Learning Translation Rules From A Bilingual Corpus’, in *NeMLaP-2: Proceedings of the Second International Conference on New Methods in Language Processing*, Ankara, Turkey, pp. 90–97; repr. as ‘Learning Translation Templates from Bilingual Translation Examples’ in Carl and Way (2003), pp. 255–286.

Collins, B.: 1998. *Example-Based Machine Translation: An Adaptation-Guided Retrieval Approach*. PhD thesis, Trinity College, Dublin.

Collins, B. and P. Cunningham: 1995. ‘A Methodology for Example Based Machine Translation’, in *CSNLP 1995: 4th Conference on the Cognitive Science of Natural Language Processing*, Dublin.

Collins, B. and P. Cunningham: 1996. ‘Adaptation-Guided Retrieval in EBMT: A Case-Based Approach to Machine Translation’, in I. Smith and B. Faltings (eds), *Advances in Case-Based Reasoning: Third European Workshop, EWCBR-96*, Berlin: Springer, pp. 91–104.

Cranias, L., H. Papageorgiou and S. Piperidis: 1997. ‘Example Retrieval from a Translation Memory’, *Natural Language Engineering* **3**:255–277.

Dennett, G.: 1995. *Translation memory: Concept, products, impact and prospects*. MSc dissertation, School of Electrical, Electronic and Information Engineering, South Bank University, London.

Esselink, B. 2000. *A Practical Guide to Localization,* 2nd ed. Amsterdam: John Benjamins.

Frederking, R., S. Nirenburg, D. Farwell, S. Helmreich, E. Hovy, K. Knight, S. Beale, C. Domashnev, D. Attardo, D. Grannes and R. Brown: 1994. 'Integrating Translations from Multiple Sources within the Pangloss Mark III Machine Translation System', in *Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas*, Columbia, Maryland, pp. 73–80.

Freibott, G.P.: 1992. 'Computer Aided Translation in an Integrated Document Production Process: Tools and Applications', in *Translating and the Computer14: Quality Standards and the Implementation of Technology in Translation,* London, pp. 15–24.

Furuse, O. and H. Iida: 1994. 'Constituent Boundary Parsing for Example-Based Machine Translation', in *COLING 94, The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp. 105–111.

Gaussier, E., J.-M. Langé and F. Meunier: 1992. 'Towards bilingual terminology', in *Proceedings of the ALLC/ACH Conference*, Oxford, pp. 121–124.

Gerloff, P.: 1987. 'Identifying the Unit of Analysis in Translation', in C. Færch and G. Kasper (eds), *Introspection in Second Language Research*, Clevedon: Multilingual Matters, pp. 135–158.

Güvenir, H. A. and I. Cicekli: 1998. 'Learning Translation Templates from Examples', *Information Systems* **23**:353–363.

Harris, B.: 1988. 'Bi-text, a new concept in translation theory', *Language Monthly* **54**:8–10.

Heyn, M. 1998: 'Translation Memories – Insights & Prospects', in L. Bowker, M. Cronin, D. Kenny and J. Pearson (eds) *Unity in Diversity? Current Trends in Translation Studies*, Manchester: St Jerome, pp. 123–136.

Hutchins, J. 1998: 'The Origins of the Translator's Workstation'. *Machine Translation*, **13**:287–307.

Jain, R., R. M. K. Sinha and A. Jain: 1995. 'Role of Examples in Translation', in *1995 IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, BC, pp. 1615–1620.

Kaji, H., Y. Kida and Y. Morimoto: 1992. 'Learning Translation Templates from Bilingual Text', in *Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, Actes du quinzième colloque international en linguistique informatique, COLING-92*, Nantes, France, pp. 672–678.

Kay, M.: 1980. The Proper Place of Men and Machines in Language Translation. Research Report CSL-80-11, Xerox PARC, Palo Alto, Calif. Reprinted in *Machine Translation* **12**:3–23 (1997).

Keck, B.: 1989. 'Theoretical Study of a Statistical Approach to Translation', TWB Technical Report, Fraunhofer Institute IAO, Stuttgart, Germany.

Kitano, H.: 1993. 'A Comprehensive and Practical Model of Memory-Based Machine Translation', in *Proceedings of the International Joint Conference on Artificial Intelligence*, Chambéry, France, pp. 1276–1282.

Kugler, M., G. Heyer, R. Kese, B. von Kleist-Retzow and G. Winkelmann: 1991. 'The Translator's Workbench: An Environment for Multi-Lingual Text Processing and Translation', in *Machine Translation Summit III Proceedings*, Washington, DC, pp. 81–83.

Langé, J.-M., É. Gaussier and B. Daille: 1997. 'Bricks and Skeletons: Some Ideas for the Near Future of MAHT', *Machine Translation* **12**:39–51.

Le-Hong, K., M. Höge and A. Hohman: 1992. 'User's Point of View of the Translator's Workbench', in *Translating and the Computer14: Quality Standards and the Implementation of Technology in Translation*, London, pp. 25–32.

Macdonald, K.: 2001. 'Improving Automatic Alignment for Translation Memory Creation', in *Translating and the Computer 23: Proceedings from the Aslib Conference held on 29 & 30 November 2001*, London, [pages not numbered].

Macklovitch, E.: 2000. 'Two Types of Translation Memory', in *Translating and the Computer 22: Proceedings from the Aslib Conference 16 & 17 November 2000*, London, [pages not numbered].

Macklovitch, E. and G. Russell: 2000. 'What's Been Forgotten in Translation Memory', in J. S. White (ed.) *Envisioning Machine Translation in the Information Future: 4th Conference of the Association for Machine Translation in the Americas, AMTA 2000*, Berlin: Springer, pp. 137–146.

Manning, C. D. and H. Schütze: 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press.

Maruyama, H. and H. Watanabe: 1992. ‘Tree Cover Search Algorithm for Example-Based Translation’, in *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation. Empiricist vs. Rationalist Methods in MT. TMI-92*, Montréal, Québec, pp. 173–184.

Matsumoto, Y., H. Ishimoto and T. Utsuro: 1993. ‘Structural Matching of Parallel Texts’, in *31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, pp. 23–30.

Matsumoto, Y. and M. Kitamura: 1995. ‘Acquisition of Translation Rules from Parallel Corpora’, in R. Mitkov and N. Nicolov (eds) *Recent Advances in Natural Language Processing: Selected Papers from RANLP’95*, Amsterdam: John Benjamins, pp. 405–416.

McTait, K.: 2001. ‘Linguistic Knowledge and Complexity in an EBMT System Based on Translation Patterns’, in *MT Summit VIII Workshop on Example-Based Machine Translation*, Santiago de Compostela, Spain, pp. 23–34; to be repr. Carl and Way (2003).

McTait, K., M. Olohan and A. Trujillo: 1999. ‘A Building Blocks Approach to Translation Memory’, in *Translating and the Computer 21: Proceedings from the Aslib Conference*, London, [pages not numbered].

Melby, A. K.: 1995. *The Possibility of Language: A Discussion of the Nature of Language*. Amsterdam: John Benjamins.

Melby, A. K.: 1998. ‘Behind the Scenes: Data-exchange standards are unsung heroes revolutionizing the language industries’, *Language International* **10**.6:30–31, 43.

Melby, A. K.: 2000. ‘Sharing of translation memory databases derived from aligned parallel text’, in Véronis (2000), pp. 347–368.

Merkel, M., B. Nilsson and L. Ahrenberg: 1994. ‘A phrase-retrieval system based on recurrence’, in *Second Annual Workshop on Very Large Corpora (WVLC2)*, Kyoto, Japan, pp. 99–108.

Nagao, M.: 1984. ‘A Framework of a Mechanical Translation between Japanese and English by Analogy Principle’, in A. Elithorn and R. Banerji (eds) *Artificial and Human Intelligence*, Amsterdam: North-Holland, pp. 173–180.

Nirenburg, S., C. Domashnev and D. J. Grannes: 1993. ‘Two Approaches to Matching in Example-Based Machine Translation’, in *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation TMI ‘93: MT in the Next Generation*, Kyoto, Japan, pp. 47–57.

Nomiyama, H.: 1992. ‘Machine Translation by Case Generalization’, in *Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, COLING-92*, Nantes, France, pp. 714–720.

O’Brien, S. 1998: ‘Practical Experience of Computer-Aided Translation Tools in the Software Localization Industry’, in L. Bowker, M. Cronin, D. Kenny and J. Pearson (eds) *Unity in Diversity? Current Trends in Translation Studies*, Manchester: St Jerome, pp. 115–122.

Pappegaaij, B. C., V. Sadler and A. P. M. Witkam (eds): 1986a. *Word Expert Semantics: An Interlingual Knowledge-Based Approach*. Dordrecht: Reidel.

Pappegaaij, B. C., V. Sadler and A. P. M. Witkam: 1986b. ‘Experiments with an MT-Directed Lexical Knowledge Bank’, in *11th International Conference on Computational Linguistics: Proceedings of Coling ’86*, Bonn, West Germany, pp. 432–434.

Planas, E.: 1999. ‘Towards Second-Generation Translation Memories’, *LISA Newsletter* **8**.2:7–11.

Planas, E. and O. Furuse: 1999. ‘Formalizing Translation Memories’, in *Machine Translation Summit VII*, Singapore, pp. 331–339; repr. in Carl and Way (2003), pp. 157–188.

Rapp, R. 2002. ‘A Part-of-Speech-Based Search Algorithm for Translation Memories’ in *LREC 2002, Third International Conference on Language Resources and Evaluation*, Las Palmas de Gran Canaria, Spain, pp. 466–472.

Riesbeck, C. and R. Schank: 1989. *Inside Case-Based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum.

Sadler, V.: 1991. ‘The Textual Knowledge Bank: Design, Construction, Applications’, in *International Workshop on Fundamental Research for the Future Generation of Natural Language Processing (FGNLP)*, Kyoto, Japan, pp. 17–32.

Sadler, V. and R. Vendelmans: 1990. ‘Pilot Implementation of a Bilingual Knowledge Bank’, in *COLING-90, Papers Presented to the 13th International Conference on Computational Linguistics*, Helsinki, Finland, Vol. 3, pp. 449–451.

Sato, S. and M. Nagao: 1990. ‘Toward Memory-Based Translation’, in *COLING-90, Papers Presented to the 13th International Conference on Computational Linguistics*, Helsinki, Finland, Vol. 3, pp. 247–252.

Schäler, R., A. Way and M. Carl: 2003. ‘EBMT in a controlled environment’, in Carl and Way (2003), pp. 83–114.

Schubert, K.: 1986. ‘Linguistic and Extra-Linguistic Knowledge: A Catalogue of Language-related Rules and their Computational Application in Machine Translation’, *Computers and Translation* **1**:125–152.

Sibley, J.: 1988. ‘Le système ALPS’, in *Actes du Séminaire internationale “La traduction assistée par ordinateur”, Perspectives technologiques, industrielles et économiques envisageables à l’horizon 1990”*, Paris, pp. 95–102.

Simard, M.: 2003. *Mémoires de traduction sous-phrastiques.* Thèse de doctorat, Université de Montréal.

Simard, M. and P. Langlais: 2001. ‘Sub-sentential Exploitation of Translation memories’, in *MT Summit VIII: Machine Translation in the Information Age*, Santiago de Compostela, Spain, pp. 335–339.

Somers, H.: 1999. ‘Review Article: Example-based Machine Translation’, *Machine Translation* **14**:113–157; revised version in Carl and Way (2003), pp. 3–57.

Somers, H., I. McLean and D. Jones: 1994. ‘Experiments in Multilingual Example-Based Generation’, in *CSNLP 1994: 3rd Conference on the Cognitive Science of Natural Language Processing*, Dublin, Ireland, [pages not numbered].

Somers, H., J. Tsujii and D. Jones: 1990. ‘Machine Translation without a Source Text’, in *COLING-90, Papers Presented to the 13th International Conference on Computational Linguistics*, Helsinki, Finland, Vol. 3, pp. 271–276.

Sumita, E. and H. Iida: 1995. ‘Heterogeneous Computing for Example-Based Translation of Spoken Language’, in *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium, pp. 273–286.

Sumita, E., H. Iida and H. Kohyama: 1990. ‘Translating with Examples: A New Approach to Machine Translation’, in *The Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, Austin, Texas, pp. 203–212.

Sumita, E. and Y. Tsutsumi: 1988. ‘A Translation Aid System Using Flexible Text Retrieval Based on Syntax-Matching’, in *Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, Proceedings Supplement*, Pittsburgh, Pennsylvania, [pages not numbered].

Svanholm, F.: 1992. ‘“The Happy Triad” – The Human, the MAT, and the MT’, in *Translating and the Computer14: Quality Standards and the Implementation of Technology in Translation*, London, pp. 15–24.

Topping, S.: 2000. ‘Sharing Translation Database Information: Considerations for developing an ethical and viable exchange of data’, *MultiLingual Computing & Technology* **11**.5:59–61.

Veale, T. and A. Way: 1997. ‘*Gaijin*: A Bootstrapping Approach to Example-Based Machine Translation’, *International Conference, Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria, pp. 239–244.

Véronis, J. (ed.): 2000. *Parallel text processing: Alignment and Use of Translation Corpora*. Dordrecht: Kluwer.

Watanabe, H.: 1994. ‘A Method for Distinguishing Exceptional and General Examples in Example-Based Transfer Systems’, in *COLING 94, The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp. 292–301.

Watanabe, H. and K. Takeda: 1998. ‘A Pattern-Based Machine Translation System Extended by Example-based Processing’, in *COLING-ACL ’98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Quebec, pp. 1369–1373.

Way, A.: 2001. ‘Translating with Examples’, in *MT Summit VIII Workshop on Example-Based Machine Translation*, Santiago de Compostela, Spain, pp. 66–80; repr. in Carl and Way (2003), pp. 443–472.

Weaver, A.: 1988. ‘Two Aspects of Interactive Machine Translation’, in M. Vasconcellos (ed.) *Technology as Translation Strategy*, Binghamton, NY: State University of New York at Binghamton (SUNY), pp. 116–123.

Wu, D.: 2000a. ‘Alignment’, in R. Dale, H. Moisl and H. Somers (eds) *Handbook of Natural Language Processing*, New York: Marcel Dekker Inc., pp. 415–458.

Wu, D.: 2000b. ‘Bracketing and aligning words and constituents in parallel text using Stochastic Inversion Transduction Grammars’, in Véronis (2000), pp. 139–167.

Zhao, G. and J. Tsujii: 1999. ‘Transfer in Experience-Guided Machine Translation’, in *Machine Translation Summit VII*, Singapore, pp. 501–508.