What's been Forgotten in Translation Memory

Elliott Macklovitch and Graham Russell

RALI, Université de Montréal
{macklovi,russell}@iro.umontreal.ca

Abstract. Although undeniably useful for the translation of certain types of repetitive document, current translation memory technology is limited by the rudimentary techniques employed for approximate matching. Such systems, moreover, incorporate no real notion of a document, since the databases that underlie them are essentially composed of isolated sentence strings. As a result, current TM products can only exploit a small portion of the knowledge residing in translators' past production. This paper examines some of the changes that will have to be implemented if the technology is to be made more widely applicable.

1 Introduction

The term "translation memory" admits of at least two different definitions, one broad and one narrow. The narrower, but more widely used, definition corresponds to the characteristics of a popular set of commercial products that includes *Translator's Workbench* from Trados, *Transit* from Star AG, *Déjà-Vu* from Atril and IBM's *TranslationManager/2*. According to this definition, a translation memory (abbreviated henceforth as TM) is a particular type of translation support tool that maintains a database of source and target-language sentence pairs, and automatically retrieves the translation of those sentences in a new text which occur in the database.

The broader definition regards TM simply as an archive of past translations, structured in such way as to promote translation reuse.[1] This definition, notice, makes no assumptions about the manner in which the archive is queried, nor about the linguistic units that are to be searched for in the archive. The narrower definition, by contrast, fixes the sentence as the privileged processing unit of TM systems and presumes automatic look-up as the privileged processing mode. It would thus exclude from the class of TMs an interactive bilingual concordancing tool like the RALI's *TransSearch* system[2], where the initiative for querying the archive resides with the user and not the system, and where any linguistic unit — full sentence, word or expression — may be submitted to the system's bi-textual database (see Macklovitch et al., 2000).

[1] This generic definition of TM is quite similar to that provided in the final report of the EAGLES Evaluation of Natural Language Processing Systems (EAGLES 1995).

[2] http://www-rali.iro.umontreal.ca/TransSearch/

While fully subscribing to Pierre Isabelle's assertion that "existing translations contain more solutions to more translation problems than any other available resource" (Isabelle et al., 1993), we contend that the current generation of commercial TM systems exploits only a small portion of the translational knowledge that resides in translators' past production. In this paper, we attempt, first, to clarify the limitations of these systems and, second, to elucidate the challenges that will have to be met in order to overcome these limitations and produce more powerful and more broadly applicable translation memories.

2 The Limitations of Current TM Systems

All the better-known commercial TM systems basically function in the same manner. A new text to be translated is first segmented into units which are generally sentences but may also include titles, headings, table cells, and other "stand-alone" elements. As the translator works his way through the new text, each successive segment is looked up in a database of past translations, or, to be more precise, a bi-textual database of aligned source and target translation units. When a match is found for a new source language (SL) segment, the system retrieves the associated target language (TL) segment from the database, which the translator may accept as is or alter as necessary. In this way, the vendors of TM systems claim, the translator need never translate the same sentence twice.

A first question that may be raised about this technology is what exactly is meant by the expression "same sentence" in this context. That is, what qualifies as an exact match between a new SL segment and the contents of the TM database? The answer is not as obvious as one might think. For example, are two SL units considered identical if they contain exactly the same wording but differ in their formatting attributes? Some TM systems discard all formatting and store only the plain text content, while others claim to offer the user the choice of whether or not to match on formatting attributes. Is a new sentence identical to a stored sentence if the wording of the two is identical except for certain non-translatables, e.g. proper names, dates or other types of numerical expressions? Trados' *Translator's Workbench* (henceforth TWB) will in fact treat the two sentences as an exact match and can, moreover, automatically replace the values of certain non-translatables in the retrieved TL sentence with the appropriate values from the new source sentence.[3] What about two SL sentences that are composed of the same lexical units, although some of these are inflected differently, say, for tense or number? In this case, few of the major TM systems will recognise the two sentence as constituting an exact match. Indeed, as Planas and Furuse (1999) point out, unless a TM system can do morphological analysis, it will have difficulty recognising that sentence (3) below is more similar to input sentence (1) than sentence (2) is:

[3] Other TM products may be able to do so as well. For the purposes of this paper, we have been able to actively experiment with TWB, which we take to be representative of the commercial state of the art. Our knowledge of other TM systems is more limited.

(1) The wild child is destroying his new toy.

(2) The wild chief is destroying his new tool.

(3) The wild children are destroying their new toy.

In particular, a system such as TWB whose notion of similarity is based on the number of shared characters (or, more generally, edit distance between strings) will conclude the contrary, since (2) differs from (1) by only 4 letters while (3) differs from (1) by 9 letters.

In a sense, such qualifications to the notion of "identical sentence" can be seen as attempts by TM developers to come to grips with a fundamental problem faced by this type of repetitions processing technology, and that is that, outside the particular context of document revisions or updates, and perhaps certain types of technical maintenance manuals, the verbatim repetition of complete sentences is relatively rare in natural language texts. Given that the overwhelming demand for translation today is not made up of revisions and updates, this imposes a serious limit on the applicability of these systems. Despite the enthusiastic welcome accorded TM technology by translators and professional translation services, one can imagine that certain users are nevertheless frustrated with existing systems precisely because of the relative rarity of full-sentence repetition in the bulk of the texts they translate, and because they are convinced, furthermore, that their archives actually contain much useful information on a sub-sentential level that is not being exploited by these systems.

Why can't existing systems retrieve repetitions below the level of the full sentence? As the discussion of examples (1)–(3) suggests, the bi-textual databases underlying these systems are composed of essentially unanalysed sentence strings. Rather than parsing a sentence into units at a finer level of granularity and attempting to align those units across the two languages, today's TM systems typically accommodate non-identical sentences within the input text by means of some notion of 'fuzzy' or approximate matching. How exactly do these fuzzy matching algorithms work? It is difficult to say with certainty because TM vendors, although they do illustrate the concept in their promotional literature and demos, do not generally provide a formal definition of the similarity coefficient that users may specify in order to constrain the search for approximate matches. Hence, it is not at all obvious just how the results of a 70% match will differ, say, from a 74% match or an 81% match. According to Planas and Furuse (1999, p. 338), "the notion of similarity ... in Trados [is] based on the number of similar characters". While this is undoubtedly true, it is not the whole story, for systems like TWB may lower the value of a match when the stored translation unit has been produced by an automatic alignment program or by a machine translation system, or when the source segment has multiple target equivalents; not to mention the opaque effects of word-order differences on the matching coefficient. Combining several distinct and incomparable factors into a single numerical measure may appear to simplify things for the user, but as a consequence users are left with a vague and ill-defined comprehension of a parameter that is central to the system.

In any event, the important point to underline is that in all cases, what these fuzzy matching algorithms are evaluating is the degree of similarity between complete sentences. When no sufficiently close match can be found for a new input sentence, current TM systems are unable to "back off" and retrieve examples of clauses or other major phrases, even though such units may well be present in the database. Allow us illustrate with a simplified, schematised example. Suppose that example (4) below is a new input sentence made up of twenty words, each five characters long. The TM database contains no exact match for (4) but does contain the SL sentence in (5). The two sentences, notice, share an identical sub-string $w_1 \ldots w_5$ which in both cases is marked off from the rest of the sentence by a comma. However, since this sub-string contains only 25% of the sentence's total number of characters, it is doubtful that any current TM system would be able to retrieve it among its fuzzy matches; for users are generally advised not to set the similarity coefficient too low, to avoid being swamped by dissimilar and irrelevant examples.

(4) $w_1\ w_2\ w_3\ w_4\ w_5,\ w_6 \ldots w_{20}.$
(5) $w_1\ w_2\ w_3\ w_4\ w_5,\ w_{21} \ldots w_{35}.$

Calculating similarity in terms of a simple character count is clearly unproductive and indeed counter-intuitive here. In the following section, we will discuss some of the strategies that could be employed by a more flexible TM system in order to reliably locate this kind of sub-sentential repetition and retrieve its stored translation. The point we want to make here is that current TM systems have little to offer in this kind of situation. The best they can do is back-pedal on the level of automation and allow the user to manually select and submit a word or phrase to the bi-textual database via a *TransSearch*-like concordancing tool.[4]

Another weakness in current TM systems that can be traced to the nature of the underlying database structure is the fact that in these systems, the very notion of a document is lost. Not only are the segmented units in a new text extracted from their context and submitted to the database in isolation, but the contents of the database are also stored as isolated sentences, with no indication of their place in the original document. As every competent translator knows, however, it is not always possible to translate a sentence in isolation; the same sentence may have to be rendered differently in different documents, or even within the same document, as Bédard (1998) convincingly argues. It is not hard to come up with examples of phenomena that are simply not amenable to translation in isolation: cross-sentence anaphora is one obvious example, but there are many others. Sceptics may argue that such problems are relatively rare, but they are missing the point. In order to evaluate a translation retrieved from memory, translators routinely need to situate that target sentence in its larger context. Current TM systems offer no straightforward of doing this because, unlike full document archiving systems, they archive isolated sentences.

[4] And even here, the graphic interface to the concordancer in TWB is such that the user can only submit a single contiguous sequence of input tokens. While this is sufficient for (4) and (5), it precludes Boolean or proximity-based searching of the kind that would be necessary to locate discontinuous translation units.

The above-mentioned article by Bédard also contains an interesting analysis of different configurations of repetition, not all of which, he maintains, warrant recourse to a TM system. In particular, if all the repetitions in a text are grouped together in a readily identifiable block, e.g. a page of introduction or the numbered clauses of a boiler-plate contract, or if the repetitions are limited to a small number of sentences, each of which recurs very often, then there may be more efficient ways to proceed than strict successive sentence-by-sentence processing.[5] Similarly, when an updated document has undergone only a few changes, it will often prove simpler to use a document comparison program to locate those source-language changes and then modify only the corresponding sentences in the previous translation rather than to resubmit the full document to TM. On the other hand, when the repetitions range over a large number of different sentences and these are dispersed unpredictably throughout the text, the type of repetitions processing that current TM products offer may well constitute the best solution.

To summarise: There is no denying the usefulness of current TM systems, particularly for texts that display a high degree of sentence-level repetition. On the other hand, existing TM systems are certainly far from optimal; in particular, their restriction to complete sentences as the sole processing unit, and their rudimentary character-based algorithms for locating approximate matches mean that these systems can exploit only a small part of the translational knowledge lying dormant in past translations. We now turn to the question of what will have be done in order develop more powerful TM technology.

3 Matching and Equivalence

3.1 Kinds of Equivalence

As we have seen, the "narrow" TM system organizes its contents on two levels: text is stored and accessed in sentence units, and similarity is evaluated principally in terms of shared characters. Neither of these is optimal from the user's perspective. Although sentences have a clear semantic basis in a way that characters do not, their variability, reflecting what some linguists think of as the "creativity of natural language", results in lower frequency rates than one might expect. Even when a sentence is composed largely of formulaic phrases, these can be combined with other material in novel and unpredictable ways and thus defeat present search mechanisms.

In this section we consider what steps might be taken to remedy this situation by extending the capacity of TM technology. The central issue is the behaviour of the system when confronted with a sentence to be translated: what criteria of

[5] Some TM systems also offer a batch mode alternative in the form of a pre-translate function. But, of course, there is no guarantee that the automatically inserted translations will be appropriate, especially if the TM database is composed of a wide variety of documents. For one translator's particularly severe assessment of this pre-translation function, see Falcone (1998).

equivalence are used in selecting candidate matches from its database, and how pertinent are these criteria for translation?

The discussion in Sec. 2 leads us to the following observation: Strict matching based on string identity between sentences yields high precision but low recall; high precision, since any result means that the entire query sentence exists verbatim in the TM and must therefore be relevant; low recall, since other relevant sentences will not be extracted owing to the low rate of verbatim repetition. The challenge is to improve the latter without significantly diminishing the former. A general approach to this problem involves establishing equivalence-classes of source-language expressions. This is the role of approximate matching; retrieving (3) as a reliable result for the query (1) implies treating *children* as equivalent to *child*. And as we saw earlier, if equivalence is defined purely in terms of edit distance it is impossible to exclude spurious matches such as that between *child* and *chief*. This section considers other more useful notions of equivalence, obtained by ignoring inflectional variation, conflating expressions of certain well-defined types, and identifying shared subsequences. These extensions require the ability to perform a more detailed analysis of source-language texts.

3.2 Inflection

One obvious step towards a useful definition of equivalence is to allow inflectional variants of a word to match (so *child* would match *children* but not *chief*). The underlying assumption here is that, for a user who wishes to find translations of a sentence containing some word w, the translation of any sentence containing an inflectional variant of w will be potentially informative, despite whatever minor adjustments need to be made to accommodate the variation.

A popular solution to an apparently similar problem in information retrieval is stemming (Frakes 1992); here, different word-forms are reduced to a common stem using little or no linguistic knowledge, sometimes yielding erratic results. Assuming that a relatively complete morphological description of the source-text language exists, more powerful models of the desired equivalences are available and seem advisable.

Let F be a mapping from inflected forms of words to their canonical base-forms.[6] What the latter are is not important here—for English, we can take the bare infinitive of verbs, singular of nouns, etc. Non-inflecting forms map to themselves. For example, $F(\texttt{lamps}) = \texttt{lamp}$, $F(\texttt{lamp}) = \texttt{lamp}$, $F(\texttt{between}) = \texttt{between}$, $F(\texttt{eaten}) = \texttt{eat}$, and so on. This is simply the basic function required for dictionary lookup in many NLP contexts, minus the various types of grammatical information that might be associated with a word-form.

The inverse of F, denoted by F^{-1}, performs the reverse mapping: $F^{-1}(\texttt{eat}) = \{\texttt{ate}, \texttt{eat}, \texttt{eaten}, \texttt{eating}, \texttt{eats}\}$. The composition of F with F^{-1} then has the effect of finding all inflectional variants of a word, using its base form as a pivot:

$$F^{-1}(F(\texttt{ate})) \;=\; F^{-1}(F(\texttt{eats})) \;=\; F^{-1}(F(\texttt{eating})) \;=\; \ldots$$

[6] Although the presentation refers to inflectional variation, the approach could be extended to deal with derivation.

$$= F^{-1}(\texttt{eat})$$
$$= \{\texttt{ate}, \texttt{eat}, \texttt{eaten}, \texttt{eating}, \texttt{eats}\} \ .$$

In the present context, we are interested less in generating all variants than in determining the equivalence of some pair of word-forms. This can be done quite straightforwardly: $x \equiv y$ iff $x \in F^{-1}(F(y))$. The obvious implementation of this technique is to compose a finite-state transducer encoding the relation F with its inverse F^{-1} (Kaplan and Kay 1994), and make matching conditional on acceptance of both strings by the resulting composite transducer.

In some cases, the classes so defined will be too inclusive. For example, many English words are categorially ambiguous: *last* as adjective, noun and verb will all be mapped by F onto the same string `last`, even though they will be translated differently. As a result, irrelevant sentences will be retrieved. This problem could be avoided by tagging the input sentence and stored source-language texts with part-of-speech information, and defining equivalence via a mapping G, like F but with category information at the pivot. However, it is a special case of the more general problem arising from multiple word-senses; in general, perfect performance in this area could only be obtained by correctly disambiguating each potentially ambiguous word.

Rather than complicating the matching and searching process, one might consider simply applying F, lemmatizing source-language texts as they are stored in the TM and the query sentence before it is looked up, so that the sentences in (1) and (3) above would both be presented to, and represented in, the system as *The wild child be destroy* . . . . However, this solution has the disadvantage of losing information; in some circumstances a user might want results ranked in a way that privileges inflectional identity and this cannot be done unless the relevant distinctions have been preserved. Both lemmatized and 'raw' text representations could be stored in a multi-level TM of the kind suggested by Planas and Furuse (1999), albeit at some cost in space.

3.3 Named Entities

A rather different kind of equivalence is displayed by sentences containing dates, times, proper names, numerical expressions, monetary amounts, etc. Such expressions tend to require special treatment, being either invariant in translation (e.g. most company and personal names) or subject to specific conversions (e.g. some place names, number formats). Moreover, they can largely be handled in a modular fashion; the treatment of a date or place-name is independent of the linguistic context in which it appears (although it may be subject to stylistic constraints), while the exact date or place-name used in a sentence has little effect on how the remainder of that sentence is translated. This property permits another refinement in TM functionality: if all possible dates are conflated into a single "archidate" representation, certain sentence pairs which are distinct when judged by edit-distance or the identity up to inflectional variance discussed in Sec. 3.2 can be treated as equivalent. The same applies to monetary amounts, names, and so on.

The Trados TWB goes some way towards this goal, recognizing some numerical and other expressions, copying or converting them automatically into the target text, and ignoring them for matching purposes. However it is not capable of handling the full range of modular expressions discussed here. Expressions of this kind are known as "named entities" in the information extraction community[7] and it is likely that techniques under development for their detection and classification could be adopted with advantage for use in more sophisticated TMs. For names in particular see Coates-Stephens (1992).

3.4 Parsing

As examples (4) and (5) in Sec. 2 illustrate, a matching criterion based on edit distance will not in general be able to retrieve relevant sentences in which even a significant subsequence is shared with the input, if the identical text forms less than a given proportion of the total. Inflectional merging and named-entity conflation may help here, but only incidentally, by altering the ratio of shared text. They are orthogonal to the the central problem, which is that the edit-distance model has no built-in conception of adjacency or constituency. Recognizing that $w_1 \ldots w_5$ form a coherent textual unit which may well be associated with one or more reusable translations is therefore beyond its ability.

Ideally, an advanced TM system would be able to analyse a source language text into units at a finer level of detail than the sentence. Since a complete parse of unrestricted input, even where feasible, is generally too expensive, techniques of shallow parsing or chunking (Ramshaw and Marcus 1995, Skut and Brants 1998) should be considered. The input sentence would then be broken into phrases or pseudo-phrases which, because they are shorter and inherently less variable, are more likely to be present in the TM than the entire sentence, and which, because they correspond to syntactically defined expressions, are more likely than a random subsequence of the same length to yield a relevant translation.

Note that this does not necessarily imply the abandonment of the sentence as the basic storage unit; only the matching criterion is changed, edit distance now playing a smaller role, if any.

4 The Suprasentential Level

In the previous section, we criticized current TM technology for its inability to provide a principled treatment of repetition at the sub-sentential level. Another area in which TM limitations are felt is their inability to process text in units longer than the sentence: paragraphs and even entire documents have a role to play in the storage and retrieval of previous translations. It is to these higher-level units that we now turn.

[7] See `http://cs.nyu.edu/cs/faculty/grishman/NEtask20.book_3.html` for a description of the MUC-6 "Named Entity Task".

A document is more than just a collection of sentences; it has global properties that are not easily associated with its lower-level components. Administrative information concerning who originally translated a certain document, when, for which client, as part of which project, who revised and approved the translation, etc. is properly treated as applying to the text as a whole rather than separately to each individual sentence. While TWB allows for similar annotations to be made at a sentential level, this cannot be regarded as more than a matter of expedience. Current TM systems provide little or no support for document management and archiving.

Even where the core functionality of a TM is concerned, namely detection of existing (partial) translations of a new document, sentence-based storage has the weakness noted in Sec. 2 of lacking a uniform treatment of extended matching passages.

A TM system which represented documents explicitly, or at least was able to reconstruct them from smaller units, would provide its users with far more flexibility than the narrow TM model permits, including the ability:

1. to search on the document-level logging information mentioned above, in order to find the most recent documents translated for this client, etc.;
2. to retrieve similar documents and their translations for use in preparatory background reading;
3. to identify and process extended passages, removing the need to treat each sentence separately;
4. to examine the context in which proposed matches for the current sentence appear.

Support for this functionality relies on full-text indexing similar to that provided by the mg system of Witten et al. (1999), or any of several commercial packages. In our view, these same functionalities need to be extended to the context of parallel documents and fully integrated with TM technology. A partial solution is adopted by the Translation Service of the European Commission, where TWB is used as a front-end to a full-fledged document management system (Theologitis 2000).

5 Conclusions

There is a certain tension between the main selling point of current TM systems ("your translations are full of repetitions — save time and money by exploiting them") and the facilities that they actually offer: most repetitions are subsentential and are difficult to locate without sorting through large numbers of irrelevant results, while others may extend over several paragraphs, making the sentence-based processing mode unnecessarily laborious.

This paper has drawn attention to some of the limitations of present TM technology and outlined a number of modifications that could be made in order to remedy them. Some of these (inflectional merging, recognition of and conflation of certain named entities) are relatively straightforward, while others (proper

name recognition, shallow parsing) are matters for continuing research. Readers may be surprised at the fact that we have made no mention of research into finer-grained alignment methods. The reason is that reliable sub-sentential alignment is less crucial for TM technology than it is, say, for example-based MT where there may be no human in the loop. The critical challenge for better TMs, in our view, is not in linking components of a source-language sentence to their target-language counterparts, but rather in finding more efficient ways of locating source-language repetitions at levels both above and below the sentence.

Current TM systems are limited for good reasons: the choice of sentence-based storage and edit distance as the main matching criterion permits efficient implementation, and makes for a program which is intuitively accessible for users. Nevertheless, we believe that there is room for more advanced TM technology, providing access to stored source texts at levels other than the sentence, and allowing more linguistically informed search. The objective, one under investigation at RALI, is something that one might call "Full Text Translation Memory".

References

Bédard, C.: Les mémoires de traduction: une tendance lourde. Circuit, **60**, 25–26 (1998).

Coates-Stephens, S.: The Analysis and Acquisition of Proper Names for Robust Text Understanding. PhD thesis, City University, London. (1992).

EAGLES Evaluation of Natural Language Processing Systems, Final Report. EAGLES document EAG-EWG-PR.2 (1995). Section E.3.1: Design and Function of Translation Memory, 140–145. Also available at `http://issco-www.unige.ch/ewg95/`.

Falcone, S.: Translation Aid Software: Four Translation Memory Programs Reviewed. Translation Journal **2(1)** (1998). `http://accurapid.com/journal/03TM2.htm`.

Frakes, W.B.: Stemming Algorithms, in Frakes, W.B. and R. Baeza-Yates (eds.) Information Retrieval: Data Structures and Algorithms, 131–160. Prentice Hall (1992).

Isabelle, P., Dymetman, M., Foster, G., Jutras, J-M., Macklovitch, E., Perrault, F., Ren, X., Simard, M.: Translation Analysis and Translation Automation. Proc. TMI'93, 201–217 (1993).

Kaplan, R.M., Kay, M.: Regular Models of Phonological Rule Systems. Computational Linguistics **20(3)**, 331–378 (1994).

Macklovitch, E., Simard, M., Langlais, P.: TransSearch: A Free Translation Memory on the World Wide Web. Proc. LREC 2000 **III**, 1201–1208 (2000).

Planas, E., Furuse, O.: Formalizing Translation Memories. Proc. MT Summit VII, 331–339 (1999).

Ramshaw, L.A., Marcus, M.P.: Text Chunking using Transformation-Based Learning. Proc. Workshop on Very Large Corpora, 82–94 (1995).

Skut, W., Brants, T.: A Maximum-Entropy Partial Parser for Unrestricted Text. Proc. Sixth Workshop on Very Large Corpora, 143–151 (1998).

Theologitis, D.: Translation Technology from Theory to Practice. Presentation at NLP2000, Patras (2000).

Witten, I.H., Moffat, A., Bell, T.C.: Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann (1999).