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

The Best of Both Worlds
- or will two mongrels ever make a pedigree

Terence Lewis
Hook & Hatton Ltd
Hook_Hatton@compuserve.com

Introduction

This paper takes a practical look at ways machine translation and translation memory applications can be combined to produce more accurate, "more human" automatic translations. The first part discusses the distinction between the two approaches; in the second half of the paper, the process employed by Hook & Hatton Ltd for combining MT and TM is discussed in detail. The paper focuses on documentation handled in a professional translation environment. It is difficult to make the decisions required in the workflow without a sound knowledge of the language pairs involved. For this reason, we will not be looking at the so-called "web translation scene" which is likely to account for a growing share of the MT market over the coming years.

Distinction between MT and TM

Whilst several MT products nowadays incorporate a "translation archive" module and some translation memory products can be linked to MT packages, it is still valid to draw a distinction between applications that attempt to analyse ("parse") and translate an (in some cases) wholly unseen text and applications that are designed to compare strings in a particular language with pairs of strings in a database and retrieve complete or partial ("fuzzy") matches according to various user-defined criteria.

The way the two pieces of text in Figure 1 are typically handled by MT and TM engines makes the distinction between the two approaches clear.

A. The Court shall have regard: a) to any benefit or compensation which that person or any person from whom he derives the title may have received or may be entitled to receive directly or indirectly from any government department in respect of the invention in question; b) to whether that person or any person from whom he derives title has in the court's opinion without reasonable cause failed to comply with a request of the department to use the invention for the services of the crown on reasonable terms[1]

B.
Het document wordt door de opdrachtgever gelezen.
Het document wordt door de Raad van Bestuur gelezen.
Het document wordt door de opdrachtgever geschreven .

Figure l

[1] Patents Act 1977, s.58(3)

For reasons whose discussion lies outside the scope of this paper, it is unlikely that sentence "A" would be correctly translated into any language by a currently available MT application. However, if this paragraph is stored in a translation memory database it will be located and completely retrieved by the TM engine. There is only a single term difference between each of the three sentences in "B"; if an MT application can translate one of these sentences correctly and all the words in the sentences are in the lexicon, it will translate all three sentences correctly. However, no translation memory program, with the exception perhaps of DéjàVu, could automatically derive the remaining two sentences from any one of these sentences stored in its database. It is paradoxically both an advantage and a disadvantage of the translation memory application that, in the current state of the art, it has virtually no linguistic intelligence.

Advantages of combining MT and TM

Figure 2 shows some of the advantages we have experienced as a result of combining machine translation and translation memory tools.

- Possibility of incorporating high-quality human translations into computer-generated documents
- Ability to harvest valuable resources from "official" bi-lingual documents in the public domain
- Potential for using "good" bilingual websites as *a* linguistic resource
- Provides a way of generating acceptable automatic translations of more complicated sentences (e.g. sentences with two or more subordinate clauses)
- Progressive reduction of requirement for MT on large projects

Figure 2

In our experience as suppliers of computer-generated translations, translation buyers prefer the "feel and look" of human translations but can be tempted by the considerable cost savings and speed offered by machine translation (with or without some post-editing). The combined MT/TM approach provides a way of increasing this "human look and feel" in automatically generated documents. To give a simple example, some translation memory applications enable the user to identify regularly recurring sentences in a document. Using the methods described later in this paper, it is possible to provide human translations of these sentences and import them into the translation memory. This exercise alone will certainly enhance the readability of automatic translations.

Figure 3 shows a typical machine-translated sentence taken from an operating manual and a likely human translation of the same source sentence, which might be stored in a translation memory.

Typical MT output	*The auxiliary machine is energised before the main system is switched on.*
Possible TM entry:	Energise the auxiliary machine *before switching on the main system.*

One of a number of approaches that we have adopted to make automatic translations "look and feel" more human is to stock the translation memory with a large number of frequently recurring sentences, such as the one shown above. This approach proved extremely useful on a recent project that involved generating automatic translations of a set of course materials on the components of the Office 97 suite. We realised out the outset that many of the recurring sentences in these course materials such as:

> *"To close all open documents without exiting the program, hold down SHIFT and click Close All on the File menu".*

had been based on text in the original Microsoft help files. Using these files as our models, we entered "human" translations of all these sentences straight into the translation memory. We knew that from then on all these "human translations" would be automatically called up by the computer as and when needed and – what is more important – would never be sent to the MT engine where they might be mistranslated.

Our fundamental principle in combining these two technologies is that the machine translation engine should only translate sentences not stored in the translation memory database, and that everything entered in translation memory is both grammatically and terminologically correct and stylistically appropriate. This means that the document has to be divided into *"known"* and *"unknown"* sentences (or segments) in such a way that only the unknown sentences can be sent to the machine translation engine. In our workflow this division is made using the various tools available within the Trados Translator's Workbench. There are other approaches and solutions.

The process

We analyse each document using the "Analyse" function in Translator's Workbench. The analysis results contain statistical information on the number of complete and partial matches between the sentences in the source document and the sentences stored in the database. On the basis of this analysis we decide how we are going to process the document. For example, a document containing less than 10% known segments will be tackled primarily as a machine translation project. On the other hand, if more than 95% of the sentences have matches in the translation memory database, Translator's Workbench will be used as the principal tool for generating an automatic translation of the source document, and MT may not even be used at all.

Figure 4 contains a screen shot of an analysis in Translator's Workbench.

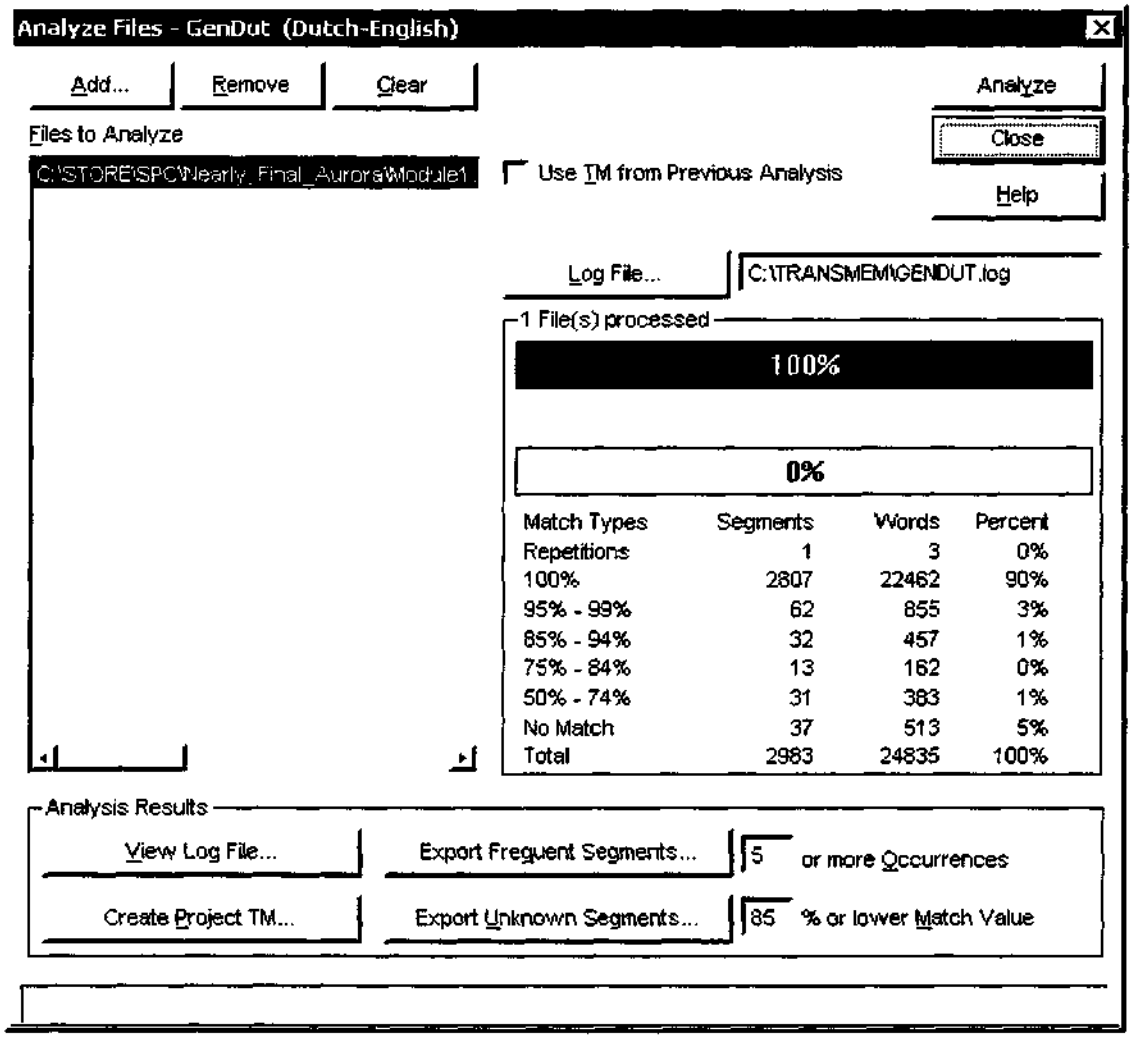

Figure 4

This analysis has shown that only 5% of the sentences (or segments) in the document are unknown, that is, they do not have any match in the translation memory database. 90% of the document will be translated automatically by the translation memory tool, so our machine translation engine will only have to deal with the unknown 5% plus the various partial matches. This analysis is typical of the results obtained towards the end of the Microsoft Office 97 course project. Such results are commonly obtained in situations where standard documents undergo periodic minor updates or revisions.

We next need to check that these unknown segments are sentences or phrases that the machine translation engine is likely to translate to an acceptable degree of accuracy. As always in machine translation, what is acceptable will depend on the purpose for which the document will be used. We perform this check by exporting the unknown segments into a text file. A simple example of such a file is shown in Figure 5.

```
<TrU>
<CrD>21091999
<CrU>LEWIS
<Seg L=NL_NL>Om de veiligheid verder te waarborgen worden alleen
gekeurd gereedschap en hulpmiddelen gebruikt.
<Seg L=EN_GB>n/a
</TrU>
<TrU>
<CrD>21091999
<CrU>LEWIS
<Seg L=NL_NL>Bovendien wordt bij alle grote projecten en
werkzaamheden met een verhoogd risico vooraf een
veiligheidsplan opgesteld, dat ook aan de opdrachtgever wordt
verstrekt.
<Seg L=EN_GB>n/a
</TrU>
```

Figure 5

Experience has taught us that our Dutch-English application is likely to translate the first sentence correctly, while the translation of the second sentence would need some post-editing. If we judge that the majority of the sentences can be acceptably translated by machine translation or would need relatively little post-editing we proceed immediately to that stage. If not, we create provide acceptable human translations of difficult sentences and import them into the translation memory database. We then send the remaining sentences off for machine translation. If we want to check the text for words not in the MT lexicon, we do so at this stage.

How users of different MT systems might handle this part of the process would depend on the MT system being used. Translator's Workbench is able to generate an "rtf' file which can be directly processed by Systran applications; it also generates an SGML file which can be handled by Logos applications. In our present work environment, we use a macro to strip out all the tags shown in Figure 5 and produce a simple text file which we then send to our MT application. When the new Java version of our program is fully implemented we expect it to process an SGML export file in the way Logos does. Figure 6 shows one of the segments from Figure 5, except that this time it is formatted so that it can be translated directly by Logos MT applications. The basic difference between the two figures is that the source text is repeated in the SGML export file; the second instance of the source text would be then replaced by the machine-translated target text. The tags will tell the MT engine what to translate and what not to translate.

```
<TrU>
<CrD>21091999
<CrU>MT!
<Seg L=NL_NL>Om de veiligheid verder te waarborgen worden alleen gekeurd
gereedschap en hulpmiddelen gebruikt.
<Seg L=EN_GB>
</LGS-EXT>
Om de veiligheid verder te waarborgen worden alleen gekeurd gereedschap
en hulpmiddelen gebruikt.
<LGS-EXT>
</TrU>
<TrU>
<CrD>21091999
<CrU>MT!
```

The next stage in the process is to handle the MT output. Logos and Systran will produce files which in format, if not in content, are ready to be imported into the translation memory database. Our own program produces a simple text file. Before this file can be imported into the translation memory database it needs to be aligned with the source text file. We previously did this with a macro written in Word 97 but have since introduced the Trados WinAlign alignment tool which produces acceptable alignment results. Figure 7 shows a couple of records in the Alignment export file which will be imported into the translation memory database.

```
<TrU>
<Quality>82
<CrU>ALIGN!
<CrD>17091999
<Seg L=NL_NL>De  doelstelling van de  COE-cursus   is  om de medewerker
in staat  te  stellen met de nieuwe werkplek  te kunnen werken.
<Seg L=EN_GB>The purpose  of   the  COE  course  is   to  enable  the
employee   to  use   the new workstation.
</TrU>
<TrU>
<Quality>90
<CrU>ALIGN!
<CrD>17091999
<Seg L=NL_NL>De  COE-cursus wordt  in 2  uitvoeringen gegeven:
<Seg L=EN_GB>Two versions  of   the  COE  course  are  provided:
</TrU>
```

Figure 7

The "Align" attribute after the creator tag tells the Trados Workbench not to accept the translation of the segment as a 100% match because it has been generated by a machine translation program. This means that as the translation memory engine processes the source file it will stop every time it reaches such a segment. We prefer to make any changes in the alignment project file and process the complete file in batch mode, so we change the "Align" attribute to something different. The amount of post-editing done in this alignment file will then depend on the success of the machine translation run and on customer's requirements. At the end of the alignment process, which generally takes a minute or so, the "alignment project" is exported and then imported into the translation memory database.

The complete document is next "translated" either by using the "Translate" option in the Translator's Workbench window. If the customer has ordered an "automatic translation", we do not look at the document any further after this stage. If the "automatic translation" is only one stage in the overall document production process, we may then send the translation to a subject specialist for revision or editing. In practice, most of our customers prefer to handle this stage themselves.

The workflow of a typical project handled by MT and TM is shown in Figure 8.

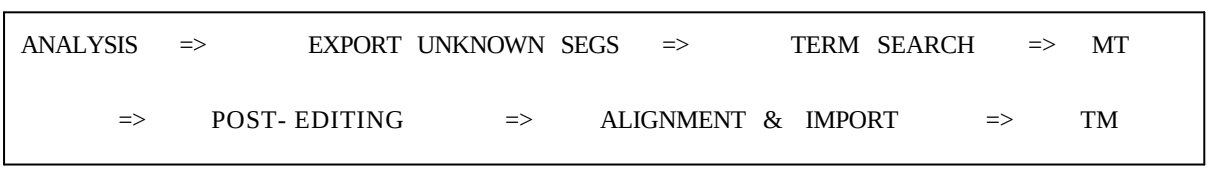

Figure 8

At first sight this might seem an extremely cumbersome process, one only really worthwhile pursuing for the translation of lengthy, repetitive technical documents, such as software training manuals or equipment operating instructions. In the case of short documents (below 2,000 words in length) it might seem just as cost-effective to have a human translator dictate (using continuous speech recognition software) straight into the translation memory application. Is this in fact the case?

To test this out, we took a document which had already been processed in Translator's Workbench and added some new text which we knew was not in the translation memory database, recreating a typical situation where an already translated document is subsequently modified or updated. The new document contained a total of 896 words. We then ran the TWB analysis on the file. It showed us that 15% of the document did not have any match in the translation memory database. We exported the unknown sentences, prepared the file for MT, machine-translated the file, carried out a quick-scan/post-edit of the machine translation output, aligned the English version with the original Dutch version, imported the alignment project into translation memory and processed the new document completely via the translation memory batch "Translate" option. The timings for this exercise are shown below in Figure 9.

DOCUMENT LENGTH:		896 WORDS
1) Analysis + Export	=	l min.
2) Preparation of file	=	l min.
3) Machine translation	<	20 sec. (15% of total doc)
4) Quick-scan/post-edit	=	l min.
5) Alignment	=	l min.
6) Import into TM	=	20 sec.
7) Translation of complete file in TM	<	l min.
TOTAL PROCESSING TIME	<	6 min.

Figure 9

Although our MT application would have processed the entire file in well under 1 minute, it would have probably taken a post-editor more than 6 minutes to achieve the quality level provided by the translation memory database, and it is unlikely that anyone could consistently dictate at a rate of over 100 words a minute for any length of time. On the other hand, the timings for items 1, 3, 5, and 6 would not change substantially if the size of the document were increased, say to 5000 words. During the test exercise carried out to establish the above timings, it took the prototype of our new Java MT application around 1 minute to process more than 6000 words of very simple sentences without even loading the core dictionary into RAM. With faster processors and cheaper RAM, speed is rarely the issue in any MT package nowadays. Such gigantic speed gains should in fact encourage developers to find ways of improving the quality of MT output by designing algorithms to enhance semantic analysis involving links to reference materials located outside the program, possibly even on the web. One hundred words a minute of near human quality translation is, in most cases, more valuable than 6000 words of re-arranged lexical transpositions.

Our own experience bears out the argument that it is cost-effective to combine these two approaches, particularly on large-scale translation projects. Technical documents contain a large amount of repetition, and as a major project progresses an increasing volume of the translation work will be handled by the translation memory engine. As I have already mentioned, towards the end of the Microsoft Office 97 project, our analyses were giving us 90 - 95% matches. Since we were making sure that all the entries in the translation memory were checked for grammar, terminology and style, we were able to deliver high-quality translations with very little effort.

This process differs from the web translation scenario where low-quality translation is acceptable in the interest of rapid communication. The model discussed here presupposes the existence of a translator or a linguistically competent user at the heart of the process. MT and TM are seen as two of many tools available. The work environment we have developed greatly resembles the translation workstation envisaged by Alan Melby more than 10 years ago. The 450 MHz PC on my desk has 383 MB of RAM. I run two MT applications, a translation memory application, pre-processing tools, post-editing tools, Multi Term and speech recognition software, and EURODICAUTOM comes after CNN on the list of Favourites in my browser. Many of you will be able to describe similar arrays of tools. Growing numbers of translators are adopting a multi-tooled approach to the translation task, using different tools for different types of document.

In this context, the exchangeability of linguistic resources is important. In the ideal work environment, it should be possible to store the MT lexicon in a terminology program such as MultiTerm, which can be accessed from a translation memory application; some MT applications do in fact allow their dictionaries to be consulted from outside the translation engine. We can access our own MT dictionary from within Translator's Workbench and are currently developing a plug-in to our new Java MT engine to enable the program to access the translation memory database directly. All that would be required would be to export the current translation memory each session - a task that takes no more than a minute. A common format such as TMX for exchanging translation memory data between tools is a right move towards complete exchangeability, and there may come a time when general and specialist translation memories become products in their own right.

Many documents are multi-disciplinary and in document production situations it may be practicable to divide up documents into parts suitable for processing through MT and parts to be handled interactively through translation memory tools. Several files may have to be combined and the translator's workbench or workstation seems the best place to do this. In this context I should like to make a passing reference to the TransRouter project.

According to the "TransRouter web site", the TransRouter project aims to build on existing translation technology and standard integration techniques in order to develop a tool which will help decision makers in translation agencies, service providers and other prospective user categories to make the most effective and appropriate use of translation technology tools and the best mix of human and computer-aided resources for a given set of documents. The tool will partly rely on data supplied by users, but will do much of its work automatically, based on computer analysis of characteristics of the text. While the technology needed to provide effective computer support for translation has been integrated into a number of commercial products such as terminology management systems, translation memories and full-scale machine translation systems, there are not many resources available on best practice in deploying them. Users may be from commercial or administrative translation services or freelance translators seeking to maximise the efficiency and accuracy of their work. Users who have to deal with large volumes of text will gain most from using the tool which will ease the decision-making process and reinforce confidence in the appropriateness of that decision. This is clearly a project we will be watching very closely.

Conclusion

The subtitle of this paper is "or will two mongrels ever make a pedigree?" Most MT and TM applications are products that were "best shots" within various technological or financial constraints at the time they were developed. DP Translator, the predecessor of Transcend allowed the user to enter a phrase in an idiom dictionary so long as its constituents were already in the core dictionary, a first step perhaps towards making MT output more human. But the technological constraints have now gone. There is no longer any real software engineering reason why an MT program cannot first mine the data in a translation memory and then pull matching translation units into a temporary file for subsequent insertion in the output file. The professional version of T1 has a translation memory feature which allows the user to align a post-edited file with the source and store the alignment project in a translation memory which can be searched by the MT engine. The developers of DéjàVu have recently incorporated in their product a function that can combine parts of two different translation units. On the other hand, the Logos Corporation's web site describes how one of the earliest developers of robust language engineering solutions is using a two-pronged approach that greatly resembles the one described in this paper. There seems to be a consensus that the MT engine should not process already translated text and that the TM tool should be intelligent enough to combine parts of several translation units into a new translation unit. One definition of "pedigree" is the "recorded purity of a breed", while the term "mongrel" conveys the idea of interbreeding or mixed ancestry and, as used in my subtitle, it conveys the notion that the mongrel is inferior. Paradoxically, however, the pedigree we are seeking should embody the best aspects of a mixed ancestry: the intelligence of the best MT software and the fidelity of retrieval of the TM tool. We do not, however, need to wait for the golden age of language engineering. The tools are already available: MT packages, TM packages, terminology management tools, controlled document tools, and editing tools. The need is, rather, for carefully devised methodologies for getting the most out of the strongest features of these tools and a well-designed interface that ensures their seamless integration and the full exchangeability of their data. Then we can have the best of all the worlds!