

# Three Perspectives on MT in the Classroom

Harold Somers

Centre for Computational Linguistics, UMIST  
PO Box 88, Manchester M60 1QD, England  
Harold.Somers@umist.ac.uk

## Abstract

This paper considers the role of translation software, especially Machine Translation (MT), in curricula for students of computational linguistics, for trainee translators and for language learners. These three sets of students have differing needs and interests, although there is some overlap between them. A brief historical view of MT in the classroom is given, including comments on the author's 25 years of experience in the field. This is followed by discussion and examples of strategies for teaching about MT and related aspects of Language Engineering and Information Technology for the three types of student.

**Keywords:** language teaching, CALL, translation studies, computational linguistics, MT evaluation

## 1 Introduction

This paper considers the teaching of Machine Translation (MT) and as such takes three different but related perspectives depending on three distinct types of students: students of computational linguistics, trainee translators and foreign-language learners. We focus particularly on MT software, i.e. software which aims to produce a translation of an input text; but the paper also covers the broad range of computer software *relevant* to the translation process, from computational tools for translators via computer-aided translation software through to fully automatic MT.

The use of MT and related software in the classroom is motivated by different concerns: one relates to teaching about computers and translation for its own sake, as part of course in one of the contributing fields such as linguistics, computational linguistics, computer science, information technology and so on. Another is teaching trainee translators and other professional linguists about translation software. A third is the role (if any) of this software for teaching languages. These are the three perspectives in the title, and they define the structure of the main part of this paper.

Our discussion will, as is normal, be preceded by a review of the literature on this subject. The literature on MT, translation software and Language Engineering in general is of course vast, but we will be concentrating on the (relatively sparse) literature on our chosen theme of MT *in the classroom*, beginning with a critical review of our own experience in this fields, and its relevance (or otherwise) to the current and future situation. Following that, Section 3 will look at the aspects of MT of interest to linguists, computational linguists, computer scientists and so on, and we will suggest some ways in which translation software can be used to illustrate these areas of interest.

In Section 4 we discuss why translators and other language professionals should know about translation software, and make some suggestions about the way the subject can be presented. We concentrate on practical aspects of familiarizing trainee translators with translation software and related themes.

In Section 5 we shift the focus to the possible use of translation software in language teaching. There is an obvious overlap inasmuch as some (though by no means all) language teaching is related to translation as a linguistic activity, but we will also review some proposals

for the use of translation software to enhance language learners' perception of contrastive differences between languages, and to help them learn aspects of second-language grammar and syntax.

## 2 A Historical Perspective

The present author has been teaching MT for 25 years. However, lest this statement be seen as self-aggrandisement let it be said immediately that, ironically, much of this experience is now felt to be largely irrelevant to the current situation, as will be explained shortly. In the early 1970s, J.C. Sager and a small number of colleagues had the foresight to set up an undergraduate degree programme at UMIST majoring in what might now be called "Language Technology". The first students were admitted in 1976 to follow a 4-year course combining Computational Linguistics (CL) with study of French or German, and later the requirement to study a foreign language became optional. The present author joined the group soon after it started.

What is noteworthy about the author's personal historical viewpoint is that for a large part of the 25 years of teaching CL, the subject has been taught in a mostly theoretical way, with an emphasis on programming, formal and mathematical linguistics, and formal linguistic theories, with discussion and consideration of applications, among which MT was always prominent. Crucially, we had little or no software to demonstrate and "play" with until very recently. Up to that time, any software that was available was generally experimental, not particularly robust or efficient, and certainly not suitable for students who did not understand well the underlying complexities. Bear in mind that PCs were not widely available until the mid-1980s. In other words, CL students might appreciate it, but translators and language learners would be distinctly unimpressed. All this has changed radically in the last few years of course, as we shall see later.

There is of course a considerable literature on MT, translation software and Language Engineering in general, but only a small proportion of that literature relates to the *teaching* of MT, or the use of MT in the classroom, and most of it is very recent. The earliest literature that we have been able to find is typified by a series of articles by Loffler-Laurian (1983, 1985, 1987) which are rather general in nature. Corness (1985, 1988) gives details of

the use of ALPS's interactive MT system with advanced learners of German. This use of MT as a type of CALL is also seen in more recent articles such as Ball (1989), Richmond (1994), Anderson (1995), and Lewis (1997), whereas other recent articles are more explicitly concerned with CAT tools and their relevance to trainee translators (e.g. Haller, 1994; Balkan et al., 1997; Bohm, 1997; Schmitt, 1998; Kenny, 1999).

### 3 MT for Students of Computing, Linguistics, and CL

Historically, MT was probably the first non-numerical use of computers proposed. From early (not entirely successful) attempts to use computers to translate natural languages grew the now well-established field of Computational Linguistics (CL). Like many other fields it has its theoretical, methodological and practical sides. We can identify, as in "general" linguistics, basic theoretical and methodological aspects applied to the various "strata" of language description that linguistics generally recognises: phonetics and phonology, orthography, morphology and word-formation, syntax and grammar, semantics and meaning, pragmatics and usage. CL focuses on computational aspects of the above, notably representation, analysis and generation. In CL we can also recognise numerous applications of these fundamental methodologies, translation being just one.

The most interesting aspect of MT for CL is that, more than any other application, translation requires "coverage" of all the linguistic levels in more than one language. For this reason MT is sometimes seen as the archetypical application of CL. Another useful feature of translation as a test-bed for CL techniques is that you can usually tell pretty well whether an MT program has "worked" (notwithstanding subtle difficulties of saying just how "good" a translation is, it is usually quite clear whether some piece of text is or is not a translation of another text).

For the student (and teacher) of CL, then, MT systems can be used to illustrate problems (and solutions) in language analysis at various levels both monolingually and contrastively. Source-text analysis requires morphological disambiguation and interpretation, word-sense disambiguation, syntactic, semantic and pragmatic disambiguation. Translation involves converting linguistic aspects of the source text into their appropriate form in the target text, thus the application of contrastive lexical and syntactic knowledge. And the generation of the target text involves the corresponding problems of style, syntax, and morphology. With the advent of spoken-language translation systems, these can be used to illustrate problems of speech processing, both analysis and synthesis. The focus of such systems on task-oriented cooperative dialogues also affords an opportunity to look at issues relating to dialogue and discourse. The multilingual aspect of these issues provides an interesting additional dimension.

Exercises can be developed to familiarize students with weaknesses and problems of MT software (these can also be used for trainee translators). Figure 1 shows some examples of "trick" sentences that we have used to show some of the subtleties of natural language and how difficult these can be for computers. Even the best MT software packages will generally have some difficulties

with some or all of the following. We show sentences for translation from French and German into English: mostly they cover the same linguistic problems, though one or two are particular to that language pair.

More generally, MT output can be used with students of CL for linguistic error analysis in general or focussing on one particular problem area, using a specially designed test suite (cf. Arnold et al., 1993). Lewis (1997) shows an example of a test suite of sentences for use with MT to investigate the translation of complex English verb forms into German. A test suite can also be used to explore the linguistic rules apparently used by the system ("reverse engineering"). Some of the examples in Figure 1 adopt this approach: for instance, the first German case explores whether the idiomatic translation *Hunger haben* → *be hungry* is maintained when the phrase is modified by an adjective in German which is rendered as an adverb in English.

More peripheral to our interests, MT offers some interesting computational problems for computer scientists, though looking at commercial software is not an especially productive way of investigating these, since it is difficult to get much information on how most commercial translation systems really work; on the other hand, translation software offers some good examples for students interested in human-computer interaction and interfaces, and especially in software documentation.

### 4 Teaching Trainee Translators about MT

It is not contentious to claim that trainee translators and other professional linguists need to understand what MT and related software can and, perhaps even more important, *cannot* do. Translators need some insight into how MT works, why it is difficult, what kind of translation tasks MT is appropriate for, what alternative computational tools are available and how to integrate them into the workflow. With trainee translators, we can usually assume some understanding and appreciation of the general problems associated with translation, though they may not have studied these in any formal way. So our aim with these students is rather to emphasize the difficulties that the computer faces, often in cases which a human translator takes in their stride.

Hands-on experience of various tools is an essential aspect of the translator's education. In the past, the expense of MT systems has made it difficult for translator-training establishments to invest in software: pricing is more oriented towards professional users, though our experience is that discounts can be negotiated with some vendors for educational establishments. More recently the cost of MT systems has fallen dramatically, and – assuming that computer labs to install the software are already available – obtaining a few systems for students to experiment with is quite a reasonable goal. We have found it useful to obtain a range of software, including systems which we know to be among the less impressive: illustrating how bad MT can be is a useful precursor to showing the best that MT can offer.

Students can of course be invited simply to familiarize themselves with the available software by doing some practice translations. Specially designed exercises which expose students to the weaknesses of the software, like the "suite" illustrated in Figure 1, can be used for this purpose.

### French examples:

1. L'oiseau entra dans la chambre. L'oiseau entra dans la chambre en sautillant.
2. Charles se suicida.
3. On a donné le livre à Paul. On a dormi dans ce lit.
4. Nous venons de finir de lire ce livre.
5. Mon cousin est beau. Ma cousine est belle. Ma cousine est riche.
6. Les pieds de la table sont très épais.
7. J'ai loué la voiture de chez Avis. Avis m'a loué la voiture.
8. Le voleur donnait un coup de pied au gendarme. Le voleur donnait des coups violents de pied et de poing au gendarme.
9. Le pilote ferme la porte. Le pilote agile le porte.
10. Vous pouvez faire des achats de votre domicile.
11. Mon ancien mari a visité une ruine ancienne.

### German examples:

1. Ich habe Hunger. Ich habe grossen Hunger.
2. Ich esse gern. Fritz spielt oft gern Tennis.
3. Das Mädchen gefällt dem Mann. Das Mädchen scheint, dem Mann zu gefallen.
4. Es wird getanzt und gegessen.
5. Hans will, dass Kurt sein Frühstück isst.
6. Ich liebe Kreuzworträtsel. Hans ist ein schneller Kreuzworträtsellöser.
7. Ulla war wegen Ladendiebstahls angeklagt.
8. Meine Armbanduhr geht vor.
9. Der ehemalige Kanzler heisst Kohl. Herr Kohl ist jetzt im Ruhestand.
10. Die Tauben lassen die Gebäude in der Stadtmitte ganz schmutzig. Die Taube hat den Olivenzweig zurückgebracht.
11. Mein Vetter ist schön. Meine Kusine ist schön. Meine Kusine ist reich.
12. In dieser Universität studieren 3 000 Studenten und Studentinnen.

Figure 1. Examples of “trick” sentences.

Other assignments and projects used by the present author with trainee translators studying MT, include the following.

Small-scale evaluation of the software: Depending on the time and effort that students are expected to put into this assignment, the evaluation can be more or less sophisticated. For most of the evaluations suggested in the literature, students have neither the time nor the resources to get statistically significant results. For example, any evaluation that requires judges to give a subjective evaluation of some aspect of the system requires quite a large experimental population. Nevertheless, they can gain a realistic impression of what is involved in setting up an evaluation even if they cannot see it through to its end result. Comparative evaluation of a single system translating different types of texts, or different systems translating the same text may be particularly revealing (e.g. Somers & Wild, 2000).

Often, students may want to work in languages for which there are as yet no commercially available systems.

In this case, a good assignment is to focus on the “for assimilation” function of MT, where it is used to produce a rough gist of an otherwise unreadable text. Students are asked to find a text (on the Web for example) in a language which is covered by the systems at their disposal but which is unfamiliar for them.

Post-editing to turn raw output into publishable quality is another exercise that students can undertake. Students should work into their native language if possible, though this of course may not always be possible. This exercise can be given as a pure post-editing exercise, or students can be asked to comment on the problem, using the given text as a case study. Students could even be asked to formulate post-editing guidelines based on a certain MT system (cf. Allen, forthcoming).

A similar exercise involves drafting controlled-language guidelines for use of a given system. Again, students should first get familiar with the behaviour of the system, and then develop a list of do's and don't's that will promote good quality translation, and avoid the main pitfalls.

An important feature of most MT systems is the ability to add items to the system dictionaries. This suggests a number of possible exercises and assignments. One way to do this is to give students a raw translation and an improved version (*not* post-edited) which is achievable by editing the system's dictionaries (this requires preparation on the part of the teacher of course), then ask the students to figure out how to edit the dictionaries so as to achieve the given target text.

More generally, students can be asked to evaluate different aspects of the dictionary-updating procedures, in particular how easy this is in general, what effect it has, and how effective it is. The (perhaps subtle) difference between these last two is that “effect” is concerned with what the details in the dictionary relate to, and the “effectiveness” is whether changing the dictionaries does actually have the intended effect. For example, one system that we are familiar with invites users to stipulate a number of “translation attributes” when entering a new noun in the dictionary, as shown in Figure 2. One could

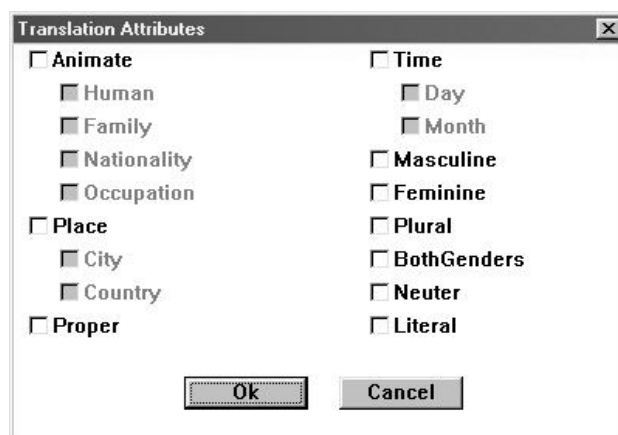


Figure 2. Semantic attributes for new dictionary entry. Screen shot from the French Assistant system, now marketed by Lernout & Hauspie

evaluate the effect of these attributes by setting up a test suite of sentences, changing particular attributes and seeing whether the translation changes. This is a kind of

“reverse engineering”, because we are trying to see how the system uses the information it asks us to give it.

Evaluating “effectiveness” tackles the problem from the other end, so to speak. In this case, we might have a certain effect in mind, and some assumption about how to achieve it. For example, in the documentation there might be some guidelines on how to get a certain result. For example, there should be a way to indicate, when entering a compound noun like (French) *poste de travail* ‘workstation’, that it is the word *poste* which should be inflected for plural. An effectiveness evaluation would confirm that marking *poste* as inflectable (and *travail* as invariant) does indeed lead to the correct translations of both singular and plural, in both directions. This is of course a trivial example, but gives some idea of the kind of exercise that can be undertaken.

We have already mentioned evaluation of software; an interesting related task is reviewing the documentation. Mowatt & Somers (2000) have developed a number of criteria for this kind of approach. They suggest evaluating whether the documentation is pitched at an appropriate level for the assumed users of the software, taking into consideration (i) the competence of the *typical user* in various areas, (ii) the competence *stated as being necessary* by the documentation and (iii) the competence *actually needed* to understand the documentation. The quality of the documentation can also be assessed by looking at the complexity of the language, the appropriateness of the jargon used, and the clarity of explanations. The completeness of the documentation can be assessed by looking to see if it explains in sufficient detail how to carry out translation itself, dictionary editing, translation memory manipulation, and any other tasks.

Hartley and Schubert (1998) emphasize the practical side of MT in the translator’s training, suggesting exercises that simulate workflow environments where MT and CAT tools might play a part. Translation of course normally involves a translator and a customer, but several other “agents” may also be involved: a reviser, a broker, publisher, and the original author. Depending on the type of translation, it may also involve specialist technicians, lexicographers and terminologists. Thinking about the “Translator’s Workstation” scenario, the technology involved includes not just computers and peripherals, but e-mail, exchange of diskettes in various formats, use of other telecommunications media, and so on, any of which may be less familiar to the student. Other aspects of the realistic translation scenario can be brought out: for translation of a manual, for example, the translator would like to have access to the hardware, or if translating a museum guide, a trip to the museum would be helpful. Many of these issues come under the heading of “localization”, a topic receiving much attention these days (Esselink, 1998).

## 5 Using MT with Language Learners

It has sometimes been suggested also that MT software can be used in the teaching of foreign languages. Obviously, inasmuch as translation is often part of foreign-language learning, we can say that learning about MT and CAT tools should be part of the curriculum for language learners. But some researchers have gone further and suggested that MT software can be used to reinforce

various aspects of the language-learning task. In this respect, the suggestion is that MT can be used as a CALL (computer-assisted language learning) tool.

The field of CALL has developed independently over the years, and there are a great number of specific computer-based tools available for language teaching. The quality, complexity and sophistication of these tools vary enormously. But MT software is generally not designed with language learners in mind, so one should be a little wary of using it for this purpose. As already mentioned, translation is an exercise that features widely in language learning curricula, and so language students should be aware of translation software. As Derek Lewis puts it,

“... language graduates need to know what the capabilities of state-of-the-art MT are and how to evaluate its role as a practical tool in the language industry.” (Lewis 1997, page 255f.)

“[F]uture employers may expect prospective graduates in modern languages to have sufficient skills and background knowledge in translation technology to influence decisions on whether or not to invest in MT.” (*ibid.*, page 261)

We have found it useful with fairly advanced students to ask them to use software to produce a first draft translation (into their native language) and then to produce an improved version (post-edited), together with a commentary. Where they have had some classes about the general difficulties and problems of MT, we ask them to relate errors in the text to problems we have discussed in class. Alternatively, we can ask them to try in their commentary to classify on a linguistic or pragmatic basis the kinds of mistakes the MT system has made.

Another, more controversial, use of MT in language learning is to use MT’s weaknesses and mistakes to bring out subtle aspects of language differences or to reinforce learners’ appreciation of both L1 and L2 grammar and style. Anderson (1995) describes use of a bidirectional English–Hebrew MT system in this way. Students manually entered sentences one by one from a suitable text corpus provided to them, noted the results, and then use native-speaker intuition and/or L2 reference works (depending on the translation direction) to identify and correct the errors. For translation into the L1, this can be a useful exercise, since the poorer-quality translations are usually too close to the lexical and syntactic structure of the source language, and this exercise can reinforce the students’ awareness of differences between the languages by showing them a bad translation into their own language. Of course, a generally low-quality translation is not of interest *per se*; rather, the text should be used (and the original source text chosen so as to bring this out) to focus on particular phrases and constructions.

On the other hand, using this technique with translations into the second language carries with it the danger of reinforcing or even introducing incorrect language habits on the part of the learner.

Richmond (1994) overcomes this problem by providing a model translation. His use of the MT system to bring language contrasts to the attention of students is somewhat idiosyncratic, but may prove to be an enjoyable exercise which “makes a change” for some students. Students are asked to type in the original (English) sentence, and note that the system gets the translation (into French in this case) wrong. They are then asked to try to modify the English sentence and retranslate it,

continuing to do so until the appropriate target text is obtained. The idiosyncratic aspect of this however is that, because the MT system he uses tends to produce rather literal translations, in order to get the desired output, the original English text has to be modified *to make it more like the French target text!* He calls this “doing it backwards”, and the pedagogic reasoning behind this is that it causes the student to focus on the differences between French and English, and to “recognize the processes by which a given meaning is expressed in French” (page 72). Anecdotal evidence from Richmond is that students enjoy the exercise and find working with the MT software challenging and worthwhile. Perhaps just from its novelty value the exercise may be worth trying.

## References

- Allen, Jeffrey (forthcoming) Post-editing. To appear in Harold Somers (ed.) *A Translator's Guide to Machine Translation*, to be published by Benjamins, Amsterdam.
- Anderson, Don D. (1995) Machine Translation as a Tool in Second Language Learning, *CALICO Journal* 13(1), 68–97.
- Arnold, Doug, Dave Moffat, Louisa Sadler, and Andrew Way (1993) Automatic test suite generation, *Machine Translation* 8, 29–38.
- Balkan, Lorna, Doug Arnold, and Louisa Sadler (1997) Tools and Techniques for Machine Translation Teaching: A Survey. Report of ELSNET-funded investigation. <http://clwww.essex.ac.uk/group/projects/MtforTeaching>.
- Bohm, Edgar (1997) A translation memory system in the university context – practical applications and didactic implications. In Eberhard Fleischmann, Wladimir Kutz and Peter A. Schmitt (eds) *Translationsdidaktik* (pp. 361–367), Tübingen: Narr.
- Corness, Patrick (1985) The ALPS computer-assisted translation system in an academic environment. In Catriona Picken (ed.), *Translating and the Computer* 7 (pp. 118–127), London: Aslib.
- Corness, Patrick (1988) MT in the University environment in 1988, In *Proceedings of AURA 1988 Conference* (pp. 47–61), Brussels.
- Esselink, Bert (1998) *A Practical Guide to Software Localization*, Amsterdam: Benjamins.
- Haller, J. (1994) Maschinelle Übersetzung in der Übersetzerausbildung. In I. Fischer (ed.) *Sprachdatenverarbeitung für Übersetzer und Dolmetscher* (pp. 207–215), Hildesheim: Olms.
- Hartley, Tony and Klaus Schubert (1998) From Testbench to Workflow: Relocating MT in Education and Training. In *Translating and the Computer 20: Proceedings of the Twentieth International Conference on Translating and the Computer* [no page numbers], London.
- Kenny, Dorothy (1999) CAT tools in an academic environment: What are they good for?, *Target* 11, 65–82.
- Lewis, Derek (1997) Machine Translation in a Modern Languages Curriculum, *Computer Assisted Language Learning* 10, 255–271.
- Loffler-Laurian, A. M. (1983) Traduction automatique et enseignement, *Revue de Phonétique Appliquée* 66–68, 87–102.
- Loffler-Laurian, Anne-Marie (1985) Informatique, traduction et enseignement des langues, *Meta* 30, 274–279.
- Loffler-Laurian, Anne-Marie (1987) La traduction automatique de textes spécialisés comme outil pédagogique, *Contrastes* 14–15, 193–216.
- Mowatt, David and Harold Somers (2000) Is MT software documentation appropriate for MT users? In John S. White (ed.) *Envisioning Machine Translation for the Information Future: 4th Conference of the Association for Machine Translation in the Americas, AMTA 2000* (pp. 223–238), Berlin: Springer.
- Richmond, Ian M. (1994) “Doing it backwards: Using translation software to teach target-language grammaticality”, *Computer Assisted Language Learning* 7, 65–78.
- Schmitt, Peter A. (1998) Computereinsatz in der Ausbildung von Übersetzern und Dolmetschern. In Mary Snell-Hornby, Hans Hönl, Paul Kußmaul and Peter A. Schmitt (eds) *Handbuch Translation* (pp. 348–350), Tübingen: Stauffenburg.
- Somers, Harold and Elizabeth Wild (2000) Evaluating Machine Translation: the Cloze procedure revisited. In *Translating and the Computer 22: Proceedings of the Twenty-second International Conference on Translating and the Computer* [no page numbers], London.