Teaching Machine Translation & Translation Technology: A Contrastive Study

Dorothy Kenny
Andy Way

Dublin City University
Dublin, Ireland

Abstract

The Machine Translation course at *Dublin City University* is taught to undergraduate students in Applied Computational Linguistics, while Computer-Assisted Translation is taught on two translator-training programmes, one undergraduate and one postgraduate. Given the differing backgrounds of these sets of students, the course material, methods of teaching and assessment all differ. We report here on our experiences of teaching these courses over a number of years, which we hope will be of interest to lecturers of similar existing courses, as well as providing a reference point for others who may be considering the introduction of such material.

1 Introduction

This paper describes the teaching of Machine Translation (MT) courses in one academic institution to two sets of students with different backgrounds. One of the authors teaches a final year undergraduate class in MT to Applied Computational Linguistics (ACL) students, while the other teaches a course in Translation Technology to postgraduate students in Translation Studies (TS), and has input into a similar undergraduate course.

Given the differing demands of both sets of students, as well as their different backgrounds, we develop a useful distinction in this paper of *users* versus *developers*. It is the case that mainstream linguists and translators may increasingly be expected to use computer assisted technology (CAT) in their jobs as translators. This may also be true of those ACL students who view their careers as more translation oriented. There is, however, an expectation that those students who take up employment as programmers or as localisation engineers will be able to design and implement new as well as existing technology in the language processing industry.

We provide a summary of the academic backgrounds of both sets of students together with a summary of their employment expectations when they leave *Dublin City University* (DCU). We then describe the MT and CAT syllabi of the two courses. Being aimed at different students on different programmes of study, there are as expected a number of components in each course which would not be considered as interchangeable between the two degrees. Nevertheless, there is some overlap between the two programmes, but even here such material is taught and assessed in different ways. We shall also address the questions of appropriate textbooks and software for each set of students. We conclude by presenting a number of dimensions along which our MT/CAT courses differ.

2 Teaching MT to Computational Linguistics Students

If nothing else, the one overriding intention of the ACL course in MT is to equip the students with sufficient background material that they may be able to talk contentfully and accurately about CAT and MT. Despite the field being relatively mature nowadays, it remains regrettable that much misinformation is still to be heard at conferences and seen on the web. The old chestnut of MT replacing translators is still heard, and unfortunately taught to students, by people who ought to know better. Translation software is as widely available now as it has ever been, but developers continue to overhype their products with misleading advertising. Integral to our positions as lecturers in the field comes the responsibility to accurately report the state of the art in the area, so that newcomers to the field—as translators, language engineers or instigators of language policy both in industry and at governmental level and beyond—do not come pre-armed with the false expectations which have harmed our area in the past. It is our job, and, we hope, the outcome of workshops such as this, to correct such false impressions and lead to an improvement in the overall perception of the area of MT. Those of us who have developed MT systems in the past and demonstrated them in various fora can only hope that the days of someone typing in a 50-word sentence consisting of strings of auxiliaries, prepositional phrases and containing ellipses, and the system either keeling over or else coming up with a hopeless 'translation' after some minutes, followed by our system tester uttering "MT is not for me!", are long gone.

2.1 Academic Background

The ACL students have a strong background in programming, language skills and good competence levels in formal linguistics and natural language processing (NLP). The degree is in its tenth year of existence and is well regarded

by the language processing industry in Ireland and elsewhere. Accordingly the students have no problems in finding appropriate employment, whether this be programming oriented or geared more towards the language competency side.

The ACL degree at *DCU* is a four year programme of study. In the first two years the students receive tuition in procedural (Java) and declarative programming (Prolog), Perl for corpus manipulation and CGI programming, and Algorithms and Data Structures. They also receive tuition in their chosen foreign language (French, German or Spanish), as well as classes in Translation and Culture and Society. On the NLP side, they are taught Logic, Maths, and Statistics, Introduction to Linguistics (Phonetics & Phonology, Syntax & Semantics, Morphology), Corpus Linguistics, Artificial Intelligence, more advanced courses on Syntax and Semantics, as well as courses on Computability and Complexity, Parsing and Statistical NLP. They spend year 3 of the degree studying at a university in a French, German or Spanish speaking country.

2.2 MT Syllabus and Assessment

The course is taught over 20 weeks, with 40 hours of lectures and 30 hours in computer labs. The primary intention, as stated above, is to provide students with a balanced view of the state of the art of MT in the new millennium. Accordingly, the first section of the course 'levels the playing field', illustrating some of these unfortunate advertising claims, and reporting on and correcting some of the popular misconceptions about MT (cf. Arnold *et al.*, 1994:6–12). To paraphrase one of Arnold *et al.*'s (*ibid*) examples, we point out that MT (and CAT) systems are best suited to repetitive material such as manuals, whereas we do not foresee there being a time where Shakespeare might be translated automatically. Given this, as well as the sheer volume of material to be translated, there is no reason why translators and MT systems cannot co-exist.

We follow on by showing that MT systems can be useful, but perhaps only if they are used in the right manner. That is, students are made aware that all systems can be expected to show an overall improvement in quality (measured in terms of accuracy and fidelity) if notions of sublanguage and controlled language are taken into account. In this context we show that merely amending the input (from passive to active, say) may cause translation quality to improve significantly. One of the students' exercises is to build a critiquing system which highlights possible problems with certain input texts (sentences too long, possible ambiguity, presence of compounds etc.), and students are then asked to rewrite the texts so that no such errors or warnings ensue while at the same time maintaining (as much as possible) the meaning of the text. Furthermore, students soon come to realise that even a linguistically impoverished, direct MT system *may* be useful in certain situations (for gisting, where we do not have sufficient competence in the source language, say).

The rest of the course is divided into two large chunks, on rule-based (RBMT) and statistical translation. With respect to RBMT, we focus on Transfer versus Interlingual approaches, examining difficult translational phenomena (such as relation-changing, or headswitching cases) as well as possible intermediate representations for such data in each system. Students write simple parsers, one with the augmented syntactic features that might be found in a transfer-based system and the other with more semantic information typical of interlingual approaches. Even for small datasets they quickly become aware of the linguistic knowledge acquisition problem that designers of RBMT systems are confronted with. On the statistical side, we show why large, good quality, representative bilingual corpora are a *sine qua non* for such approaches, and hypothesize ways in which such corpora might be extracted automatically from the Web. We then look in some detail at the major alignment algorithms which have been published (Brown *et al.*, 1991, Gale & Church, 1993; Kay & Röscheisen, 1993), prior to looking at Translation Memory (TM), Example-based MT (EBMT, cf. Somers, 1999) and other approaches to statistical MT (Brown *et al.*, 1990).

Each 4th year ACL student is required to produce a substantial NLP project in an area of their choosing. Many of the students opt for an MT project. Some examples of this year's projects include:

- Complex Lexical Transfer in Machine Translation;
- On-line Dialogue Translation for Monolingual Users;
- Using Data-Oriented Parsing for a Statistics-based Machine Translation System;
- Translator for Spanish to English Weather Reports;
- Development of an Example-based Machine Translation Tool;
- A Machine Translation System for Recipes.

It can be seen, therefore, that the theoretical background provided to these students in the course on MT enables them to write sizeable MT systems of reasonable complexity. It is for this reason that we view such students primarily as developers of MT systems.

3 Teaching MT/CAT to Translators

Dublin City University has two translator-training programmes, one undergraduate and one postgraduate. Undergraduate students take a core module in translation technology in fourth year. This module covers 24 contact hours, all delivered in computer labs. Postgraduate students take a double module (48 contact hours) in translation technology, again in computer labs, and delivered at the beginning of their one-year taught course. Postgraduates get more contact hours for two reasons: firstly we assume that they have already reached a certain level of competence in their non-native languages, and no longer need formal instruction in these languages. This frees up time to concentrate on issues that relate more immediately to the working life of a translator. Secondly we cannot assume that postgraduates have the same level of experience in using computers as undergraduates, and typically spend the first couple of weeks

getting postgraduates up to scratch on operating systems, file management, character handling, etc. The bulk of the postgraduate course covers CAT, especially TM and MT. From the next academic year onwards more time will be devoted to translation workflows (cf. Sprung, 2000), and a more in-depth look at MT, including use of controlled language.

3.1 Academic Background

By the time translation students come to study CAT/MT they can be expected to be familiar with the Windows operating system (but not necessarily any other OS), and to have mastered the basics of file management and Word for Windows. Translation students typically have no background in mathematics or statistics, no programming experience, and little or no training in formal or computational linguistics. They can, however, be expected to have excellent command of their source and target languages, and to have the transfer skills required to translate between the two. They are normally well practised researchers, used to getting up to speed in the intricacies of the specialised areas in which they have to translate. They are also alert to nuance, the importance of cohesion and thematic structure in creating texture, and the roles that textual function, target language audience and text type might play in making low and high-level translation decisions.

All the above-mentioned (non-computer) skills are highly valued in (human) translation circles, and both linguistic and computer skills can usefully be transferred to CAT, but they may not necessarily be adequate in scenarios involving MT. In fact, some commentators would argue that the traditional values instilled in student translators are somehow at odds with the requirements of workplaces where MT is the norm (cf. Schäler, 1998). While it is true then that the use of CAT tools such as TM relies to a certain extent on translators extending their traditional skills without having to rethink their traditional values, this does not apply to the use of MT by translators. Having said that, the use of CAT tools does raise some interesting questions relating to translators' self-image and remuneration, questions that should not be ignored in translator training.

3.2 CAT Syllabus and Assessment

Students are introduced to the basic concept of TM in a practical session where they use *Trados's Translators Workbench* to translate two short texts, typically an excerpt from promotional material for a new software release, and an 'update' of the same text, with minor or major adjustments made in a number of sentences. Students first create a TM database to store source and target segments as they proceed through the translation job. As the TM database is initially empty, and the potential for matching between it and the source text accordingly limited, the pedagogical focus in translating the first text is on source text segmentation. Students learn by induction how the software segments source texts, and also how to override erroneous segmentation decisions made by the system. (Source texts are chosen deliberately to create the potential for such segmentation errors.) In the second translation job, students see the benefit of having committed their previous job to memory, as the TM begins to throw up 100% and fuzzy matches with the updated source text. As the system unexpectedly—and inevitably, given the default settings and the particular source text—fails to give students a fuzzy match for a segment which is very like one they translated in the previous texts, students also learn about fuzzy match thresholds, and how to change default settings. In subsequent lab hours students learn how to commit legacy material to memory using *Trados's WinAlign* tool. Here students become aware of the important role played in automatic alignment by sentence length, and paragraph and sentence progression.

In their assessment, students continue to learn by doing. The translation task is scaled up to embrace some thirty short source texts, and students are asked first to find out how useful translation memory technology will be in completing the translation task. This allows us to introduce source text analysis tools, and to teach students to distinguish between repeated segments within texts or families of texts (known as 'repetitions' in Trados), and matches between memory and source texts. The initial results from the analysis tool in Trados show the texts to be not quite as repetitive as one would have expected. Once the students have translated an initial batch of usually five texts, they then run the analysis tool again to see whether they get fuzzy matches between the remaining untranslated source texts and the segments they have already committed to memory. Again the results are disappointing, and students' attention starts to focus on why this might be the case. The answer, of course, is that texts that appear repetitive to human beings because their semantic content is highly repetitive—as is the case with the weather forecasts used in this exercise—might not be formally repetitive, given the fact that human beings do not necessarily say the same thing in the same way all the time. The final step in the assessment involves getting the students to write guidelines for meteorologists on how to write TM-friendly weather forecasts. This part of the exercise makes the students reflect on issues of controlled language and translatability, and do so from the proactive vantage point of one who wishes to initiate the use of a certain technology in the translation process. It also drives home the point that translation memories revolve around source texts and monolingual matching, features that tend to surprise even seasoned translators who are nonetheless new to this technology.

3.3 Non-technical Issues: Working Conditions and Remuneration

The use of source text analysis tools in CAT also raises the thorny question of remuneration. Although there are other less mercantile reasons for using translation memories (cf. O'Brien, 1998; Heyn, 1998), clients who request translators to use this technology often do so in an effort to cut costs. They argue that if source texts are repetitive, and if much of the translation can simply be pulled out of memory, then translators are not entitled to receive the full going word or line rate for their work. Prices for translation jobs are thus beaten down on the basis of the number of repetitions in the source text and of 100% matches and fuzzy matches in

memory. This approach to pricing translations raises interesting ethical questions for translator trainers. We want our students to have a good command of technologies currently used by translators, but why should we encourage them to reduce their own income, especially given the investment they will have had to make in hardware, software, and their own training? We approach this issue by telling our students how professional translators deal with the prospect of reduced payment for their work: some refuse outright to offer any discount based on match type, although they may offer an hourly rate for translation completed using a translation memory; others comply with the client's wishes, if the client is particularly valued, or the translator for some reason feels she/he has no choice in the matter. In recounting the experience of professional translators, we draw on our personal experience of translation bureaux and agencies, our contacts with professional organisations like the *Irish Translators' Association* and the *Institute of Translation and Interpreting* in the UK, and product reviews published by well-known translators and software critics like Michael Benis.[1]

Benis's reviews are particularly instructive for other reasons: they emphasise the importance of good ergonomic design of software in ensuring a comfortable and injury-minimising work environment for translators; and they contrast the translation memory products of a number of different software companies, introducing an element of competition between products that is difficult to orchestrate within the confines of a short module on a busy academic programme.

3.4 MT Syllabus and Assessment

MT also features on our translator-training programmes. Because of constraints imposed by time, the availability and cost of MT systems, and students' prior knowledge, we concentrate on one PC-based system, *Globalink's Telegraph*. Our (1996) version of this product allows the user to set certain translation options (such as choice of tenses) and adaptation of lexica is possible. In addition, *Altavista's* implementation of *Systran*, which can be accessed over the web, is widely used. (A further constraint that applies to MT, but not to the use of translation memory, is the limited or non-availability of certain language pairs. There is no commercially available MT system that handles English/Irish translation, for example.) Again the emphasis is on learning by doing. Both *Telegraph* and *Babelfish* provide reasonable translations, but enough errors are introduced into the translation process by both systems that discussion points arise on the nature of the role of the translator in the translation process, the text type to be used and the whole issue of automation and pre-/post-editing. Students are typically asked to develop a suite of sentences to test the software's performance given particular syntactic structures, lexical ambiguities, etc. Later, in their assignment, students can choose to run a number of different texts through each system and comment on how the system treats the same and other discourse phenomena. Again students are asked to reflect on how changing the input text can facilitate translation by machine. Once they have experience of what MT can and cannot do, students can reflect in a more realistic and less defensive way on the role of machines in the translation process.

4 Comparison of the two Courses

It is our contention that if MT and CAT tools are to be used productively, then students need to be informed as to precisely how and why they ought to be used. Accordingly both of us spend a sizeable portion of our respective courses making this clear, focussing on input strategies involving notions of controlled language and sublanguage. This may be exemplified by using systems which produce relatively poor output.

While the ultimate goal of the ACL students is high quality output from MT systems, the TS students are more interested in CAT, and especially TM tools. This can be contrasted with the ACL students who spend more time on EBMT. While both EBMT and TM require aligned corpora, the TS students use the built-in alignment tool *WinAlign*, whereas the ACL students may be expected to write their own alignment software.

Regarding knowledge representation issues, in RBMT systems the ACL students question how lexica and large rule sets may be developed, while TS students focus more on TM examples, where knowledge of texts is paramount.

In sum, the ACL students are concerned with *automation*, while the primary focus for the TS students is *user interaction* with CAT tools.

4.1 Textbooks

We provide here a short commentary of the appropriateness of the major textbooks in the area. It is unfortunate that Arnold *et al.* (1994) is out of print[2]. As stated above, we find the introductory sections of this volume to be useful for both sets of students. Hutchins & Somers (1992) is now ten years old, and while much of the background material remains an excellent introduction to many of the major themes, others (especially the case studies presented) are now rather dated. Interestingly perhaps, both of us use different sections of this book given the different students with whom we are confronted. The background material on linguistics and computational issues are ignored in teaching MT to the ACL students, given their prior exposure to such material in previous years. In contrast, these sections are useful to TS students, who have no in-depth formal training in either of these areas. The core chapters of the book, which centre on basic strategies, are presented to the ACL students to a deeper level than the TS students receive, because this material is more important for developers. The first three chapters of Trujillo (1999) provide essential background to the TS students, which the ACL students have received elsewhere. In contrast, the technical discussion and Prolog examples provided in chapter 6 are excellent material for this latter group, whereas this would be rather daunting for the TS students.

[1] Many of which are available from the *Atril*-sponsored website at `http://www.transref.org/`.

[2] An electronic copy is available at `http://clwww.essex.ac.uk/MTbook/`.

5 Concluding Remarks

In conclusion, we note that the difference between teaching MT/CAT to translators and computational linguists manifests itself in at least three different ways:

- We concentrate on learning by induction in the case of the translators whereas deduction is the starting point for computational linguists;
- We focus on how technology affects working conditions, pay and professional self-image in the case of translators, and on design and technical implementation issues for computational linguists;
- Commercially available products are more important for translators whereas computational linguists typically focus on experimental systems.

While we teach two different courses to two different sets of students, and while we have found it useful to differentiate between users and developers, it should be stressed that each set of students is aware of how its core competencies affect and indeed complement each other's. While they are discrete groups within *DCU*, they are likely to interact in professional environments after graduating. Finally, we trust that our experiences will be of interest to lecturers of similar existing courses, as well as providing a reference point for others who may be considering the introduction of such material.

References

[1] Arnold, D.J., L. Balkan, R.L. Humphreys, S. Meijer and L. Sadler (1994): *Machine Translation: An Introductory Guide* Cambridge, Ma./Oxford: Blackwell.

[2] Bowker, L., M. Cronin, D. Kenny, and J. Pearson (eds) (1998): *Unity in Diversity? Current Issues in Translation Studies*, Manchester: St. Jerome.

[3] Brown, P., J. Cocke, S. Della Pietra, F. Jelinek, V. Della Pietra, J. Lafferty, R. Mercer and P. Rossin (1990): 'A Statistical Approach to Machine Translation', *Computational Linguistics* **16**:79–85.

[4] Brown, P.F., J.C. Lai and R.L. Mercer (1991): 'Aligning sentences in parallel corpora', in *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*, University of California, Berkeley, pp.169–176.

[5] Gale, W.A. and K. Church (1993): 'A program for aligning sentences in bilingual corpora', in *Computational Linguistics* **19**(1):75–102.

[6] Heyn, M. (1998): 'Translation Memories: Insights and Prospects', in Bowker *et al*. (eds), 123-136.

[7] Hutchins, J.W. and H.L. Somers (1992): *An Introduction to Machine Translation*, London/San Diego: Academic Press.

[8] Kay, M. and M. Röscheisen (1993): 'Text-translation alignment', in *Computational Linguistics* **19**(1):121–142.

[9] O'Brien, S. (1998): 'Practical Experience of Computer-Aided Translation Tools in the Software Localization Industry', in Bowker *et al*. (eds), 115-122.

[10] Schäler, R. (1998): 'The Problem with Machine Translation', in Bowker et al. (eds), 151-156.

[11] Somers, H.L. (1999): 'Review Article: Example-based Machine Translation', *Machine Translation* **14**(2):113–157.

[12] Sprung, R. (ed) (2000): *Translating into Success: Cutting-edge strategies for going multilingual in a global age*, Amsterdam/Philadelphia: John Benjamins Publishing Co.

[13] Trujillo, A. (1999): *Translation Engines: Techniques for Machine Translation*, London: Springer.