MT SUMMIT IX

WORKSHOP ON

TEACHING TRANSLATION TECHNOLOGIES AND TOOLS (T^4)

(Third Workshop on Teaching Machine Translation)

in conjunction with the
Ninth Machine Translation Summit

New Orleans, USA
September 27, 2003

Association for Machine Translation in the Americas

FOREWORD

Machine translation and computer-aided translation have become key technologies in the present-day globalised communications scene. These are truly cross-disciplinary technologies which should not be used without a certain level of understanding. As a result, many universities and academic institutions teach courses on MT and related language technologies, both at graduate and at undergraduate levels. Courses may be aimed on the one hand at translation and linguistics majors, or on the other at computer science or software engineering majors, or may even be part of the curriculum for learners of a foreign language (cf. Kenny & Way, 2001; Somers, 2001). Translation technologies are of interest also to students involved with information technologies in a multilingual setting (electronic commerce, localization, multilingual documentation, cross-language information retrieval). The cross-disciplinary nature of the topic poses important challenges to instructors, who have to deal either with computer science and software engineering students lacking an adequate background in linguistics and translation or with translation and linguistics majors with little experience of computer science. The existence of widespread preconceptions, fears, and expectations about MT, which interfere with the learning process, also needs addressing by instructors.

On the other hand, due to the growth of the internet, both commercial and experimental MT systems are more readily available than ever, and, if appropriately used, may be very beneficial for education. The World-Wide Web may also be seen as the medium used for teaching, as it may naturally integrate real MT systems as part of the learning environment.

This "Teaching Translation Technologies and Tools" workshop is the third in a series of workshops addressing the teaching of MT and other translation technologies; its predecessors were the "Teaching Machine Translation" workshop at MT Summit VIII in Santiago de Compostela (Spain) in September 2001 (`http://www.dlsi.ua.es/tmt/`, proceedings online) and, more recently, the 6th EAMT Workshop "Teaching Machine Translation" held in Manchester in November 2002 (`http://www.ccl.umist.ac.uk/events/eamt-bcs/`). As at its predecessors, the workshop will give attendees the opportunity to know about other instructors' approaches to teaching translation technologies and tools and to discuss their views. It is indeed the case that many instructors are faced with the task of planning or teaching a related subject, and need guidance on how to organize their syllabus or plan their lectures or laboratory assignments. As with the previous workshops, we hope this workshop will give them the chance to obtain useful suggestions to perform the task.

The proceedings of the workshop include eight papers. Even if the call for papers allowed for a broader scope including computer-aided translation, all but one paper addresses the teaching of MT proper.

Three papers describe complete courses. Teruko Mitamura, Eric Nyberg and Robert Frederking describe in detail an MT course aimed at graduate students. Mikel Forcada's paper describes how to teach a "Computers in Translation" course to undergraduate translation students in limited time. Andy Way and Nano Gough focus on the teaching of empirical (example-based, statistical) approaches to MT.

The remaining papers describe more concrete aspects of the teaching of MT in a variety of academic environments. Judith Belam's paper addresses teaching of post-editing to students in real environments and to a rather unusual audience (language learners) and argues in favour of using MT as a tool to teach foreign languages. Benoît Robichaud and Marie-Claude L'Homme describe how the automation of translation may be incrementally taught using a carefully designed progression of laboratory assignments. Cristina Vertan and Walther von Hahn describe the teaching of the software engineering aspects (specification and evaluation) of MT during the implementation of toy systems. Toy systems written in Prolog are used by Harold Somers to illustrate and teach various classical approaches to MT. Kevin Knight describes a number of resources which can be used for the introduction of the concepts and techniques underpinning Statistical MT.

The organizers hope that the selection of papers presented here will be of interest to a broad audience of practising and future instructors of MT and computer-aided translation.

References

Kenny, Dorothy and Andy Way (2001). 'Teaching Machine Translation & Translation Technology: A Contrastive Study', in *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, Spain, pages 13—17.

Somers, Harold (2001). 'Three Perspectives on MT in the Classroom', in *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, Spain, pages 25—29.

Workshop Organisers

Mikel L. Forcada, Universitat d'Alacant, Spain (mlf@dlsi.ua.es)
Harold Somers, UMIST, Manchester, UK (harold.somers@umist.ac.uk)
Andy Way, Dublin City University, Ireland (away@computing.dcu.ie)

Teaching Translation Technology and Tools (T4)

A full-day workshop at Machine Translation Summit IX

September 27th, 2003, New Orleans, Louisiana

Table of Contents

"Buying up to falling down": a deductive approach to teaching post-editing

Judith Belam

University of Exeter
Foreign Language Centre
Exeter
United Kingdom
J.M.Belam@ex.ac.uk

Abstract

A course in machine-assisted translation at final-year undergraduate level is the subject of the paper. The course includes a workshop session during which students compile a list of post-editing guidelines to make a text suitable for use in a clearly defined situation, and the paper describes this workshop and considers its place in the course and its future development. Issues of teaching MT to language learners are discussed.

1 Introduction

Our course in machine-assisted translation at the University of Exeter is now about to start its third year. It is for final-year undergraduates, about to take their first degree in modern languages. The MT course forms an option in final year, worth 15 credits of a 120-credit course.

It seems to be unusual to teach MT to language learners at this level. They have no background in computing and are not trainee translators or even postgraduates in translation. Some would even object to exposing language learners to the incorrect models which form an inevitable part of the study of MT. I do not believe, however, that this drawback outweighs the possible advantages (Belam, 2002). Some examples of teaching MT to language learners, and its possible benefits, are discussed by Somers (2001). Derek Lewis also pointed out how an MT course can be of benefit to language learners: "many students have expressed the view that they have increased their cognitive knowledge of German grammar through having to enter information in the system's dictionaries; for those students whose command of formal grammar is weak, the MT dictionaries appear to provide a stimulus for researching areas of basic grammatical structure." (Lewis, 1997). In this paper I would like to explore these benefits further by showing how the inclusion of post-editing in an MT course is appropriate for language learners. It can be shown to help their language learning and their translation skills at an appropriate level, and it also helps their awareness of the communicative aspects of language, and gives them a perspective on the use of foreign languages in the workplace.

2 The place of post-editing in the MT course

Post-editing has been part of the course from the beginning but it was included at first rather as a supplementary element than a core part of the syllabus. The course aimed mainly at teaching students how to handle the software, and concentrated on how to get the best from MT raw output, with sections on choosing appropriate materials to translate, pre-editing and creation of dictionaries to improve the raw output. There was a tendency to look upon post-editing as a kind of last resort, the final solution for any problem that could not be solved in other ways; almost like cheating. After devising ever more sophisticated pre-editing rules and adding words and phrases to the dictionary, any remaining problems just had to be dealt with by post-editing. It was almost an admission of defeat.

It does deserve, however, to be treated as a topic in its own right. Sharon O'Brien has shown the importance of teaching post-editing to translators (O'Brien 2002), underlining not only its importance in the translation workflow process but also the fact that it is a separate skill which needs to be acquired separately. We also found this year that students were particularly interested in it and requested that extra time should be spent on it.

This is partly because the group doing this MT course seems to have a tendency to develop collective enthusiasms. In the first year they were all interested in syntactic issues and pre-editing; this year post-editing was the theme. I think, however, that there were two other reasons as well. Firstly they are asked to do an evaluation of an MT system as assessed project work, and time and effort spent post-editing is a good concrete measure of translation quality. Secondly, post-editing brings the human versus machine translation comparison into sharp focus. If you want to know exactly what the gap is between MT output and human translation, you cannot do better than to set out to actually bridge that gap by post-editing.

3 Post-editing in class

We have extremely limited time to do our course, as we have to fit it into one semester. There is one lecture on post-editing, organised around some basic principles of post-editing and how to measure how much you do. The required pre-course reading is Jeffrey Allen (2003) and also recommended is Hans Krings' *On repairing texts.* (Krings 2001). Students are familiarised with the distinction between rapid, minimal and full post-editing. As defined by Allen, a rapid post-edit is "a strictly minimal editing on texts in order to remove blatant and significant errors … stylistic issues should not be considered". The full post-edit, by contrast, aims to produce a text which is of the same quality as a professional human translation. Anything in between is classified as a minimal post-edit, which Allen describes as "a fuzzy, wide-ranging category". In a minimal post-edit, a text is edited with a specific readership in mind. The amount of editing may be quite extensive if the text is required to be of fairly high quality, or quite limited if the text's usefulness is likely to be short-lived, or if the information is more important than the style of delivery.

The students then have a practical assignment to do which involves post-editing a text and commenting on the process, and it was at this stage that students began to ask for more precise guidelines. There is a reluctance even at this level to accept that there is no one right answer, one accepted method of doing things, and they wanted to know exactly what constituted a good rapid post-edit, exactly how much more was enough for a minimal post-edit, and so forth. While it seems to be true that there are no precise and generally accepted guidelines for various types of post-edit, it was obviously going to be useful to have a set of rules to which they could work, so we organised a discussion workshop to try to devise an operational set of our own. We concentrated on the rapid and minimal post-edits, as the full post-edit was assumed to be indistinguishable from a full high-quality translation.

We work from German into English as this is the language direction which happens to be common to all students. In the workshop I gave them a text with its raw translation, and in small groups they started the post-edit, in order to arrive at general rules from the particular cases they encountered. For this reason the text had to be longer than the ones we normally use for other aspects of MT. The texts are attached at the end of this paper and paragraph references given below refer to these texts.

Of course they rapidly arrived at the most uncontroversial changes which would have to be made in any post-edit (I do not know if Jeff Allen invented the phrase "blatant and significant errors" but I find it particularly apt). So clearly *Himmlischens* must change to *heavenly* or *divine* (paragraph 2) and the rule is easily formulated: *"Correct any word which has not been translated."* Equally obviously *Annuitant carriage* must change to *reindeer sleigh* (paragraph 2) giving rise to the rule: *"Retranslate any word which has been incorrectly translated"* and so forth. As the students started to approach less obvious errors, however, care needed to be taken, as there was a tendency for the post-editing rules to become too detailed. They thus became unwieldy and hard to understand, and applied to so few cases that they became useless. So I had to work to keep the rules simple and generally applicable. For example, in the case of *it became coldly > it became cold* (paragraph 5) this came under the general heading of *"incorrectly translated"* rather than trying to make up something like: *"Correct any adjectives which have been wrongly translated as adverbs"*.

The next problem which arose was that the students had a tendency to want to correct everything, and needed to be reminded that if you are going to do a complete post-edit, you might as

well translate from scratch in the first place. So in order to arrive at a minimum set of rules for a rapid post-edit, we started to refine the rules along two lines:

(i) bearing in mind how long a correction will take

(ii) bearing in mind whether a correction is necessary.

In this way I could demonstrate that we would need to refine even the most basic and obvious rules: *"Correct any word which has not been translated"* became *"Correct any word which has not been translated and does not resemble its TL equivalent"* when we considered that it was not necessary, for understanding, to change words like *Prozedere = procedure, Choraele = choir,* or *Shoppen = shopping.* Equally the rule *"retranslate any word which has been incorrectly translated"* became: *"retranslate any word which has been translated so wrongly as to make the sentence meaningless"* to take account of the fact that it is really unnecessary to change *purchase intoxication* to *shopping fever* (paragraph 4), even though the expression would not be considered acceptable in a full translation.

We then spent some time discussing the extent to which the principle of saving time by making only necessary corrections could sometimes be sacrificed to practicality. If a correction could be made with little expense of time or effort, you might as well make it. So, for example, *Grapes/clusters* can change to *clusters* (paragraph 3). A big gain in terms of text quality for very little effort. On the other hand, I had to emphasise that the time constraint is absolute, even if you have to do things sometimes against your instincts. It took students a little while to accept such controversial rules as: *"Do not tidy up the style"*. For example the phrase *"von unten rechts nach oben links",* (paragraph 2) describing the direction Santa Claus' sleigh takes in the neon light display, is translated *"from down right to upward left"*. While stylistically unacceptable, this is not only perfectly clear but also hard to correct. How would you explain that in clear concise English? *"From the bottom right-hand corner he rises diagonally to the left .."* (too clumsy) *"from bottom right to top left.."* (too mathematical) *"he rises steeply into the sky.."* (too unfaithful to the original) … some minutes of discussion on the subject of course persuaded students that we had all put in far too much cognitive effort to make the correction efficient in terms of time spent! A similar rule along these lines was: "*Do not spend time deciding on the very best word or phrase*". Thus the phrase *"which met it deeply in the heart"*(paragraph 3) for *"which struck to the very heart of the city"* would remain in spite of the curious image it conjured up. More controversial still was my suggestion: *"Delete anything which would take time to correct and does not add significantly to the meaning"*. Following this rule I proposed to delete the phrase, *"buying up to falling down"* (paragraph 4). It is not necessary anyway, being a retranslation of the German translation for the expression *Shop until you drop.* But it is also distracting for two reasons: firstly it takes time to work out why the phrase is there, and that the English phrase appears in the German original; and secondly, it is a rather amusing – I almost said clever – paraphrase of the original, and one could imagine spending some time discussing it. But it took a long time to persuade these students, who are doing classical translation courses in parallel to the MT course, that anything could justify simply leaving out a section of the text, given that they would be severely penalised for doing this under any other circumstances.

It proved relatively easy, however, to formulate rules for a rapid post-edit based on correcting the most obvious errors as quickly as possible. More disagreement arose over the second constraint, correcting only what was necessary, as there was no clear consensus about what was necessary. It was impossible to formulate guidelines for a minimal post-edit without asking the question "necessary for what"? In the end we found that the only way to get to any kind of consensus, or even realistic discussion, we had to imagine a very precise situation where the translation would be used. Still using the same text about Christmas shopping, we therefore devised a scenario in which our client, the user of the translation, is working for a business in New York and wishes to know whether the efforts to attract the overseas Christmas shoppers were being successful. In particular, how is the pre-Christmas New York shopping experience viewed by the Germans? Pressed for time in the busy run-up to the season, the client wants the article to be translated as quickly as possible in order to take it to a meeting of fellow retailers, to show that the German tourist

has a positive attitude towards the pre-Christmas shopping experience.

Immediately the whole debate about acceptability fell into sharp focus. According to this scenario, the client is in a business meeting so he does not want anything in the text which is going to waste time or be distracting. So we must add two more rules: *"Correct anything which may be comprehensible, but which takes time to work out"*. According to this rule *"the economic situation is no more crucial brake" > "the economic situation is not significantly slowing consumption"* (paragraph 3). Secondly, *"Correct anything which is startling or disconcerting"*. According to this rule *"gloss"* changes to *"sparkle"*(paragraph 1), and *"seduction"* to *"temptation"*(paragraph 7) in order to avoid discussion about the inappropriate negative connotations of *seduction* in this context.

In this imaginary meeting, the required translation quality demands something between the rapid and full post-edit. The client is with his fellow-retailers: they are normally, and especially at Christmas, his competitors, so he does not want to appear incompetent or silly, but for the purposes of this meeting, he is discussing a common problem with them, so he does not need to impress or persuade them. He has chosen the text and presented it to the meeting, so he is anxious that it should not sound stupid or ignorant, but equally it is clear that he has not authored the text and he is taking no responsibility for it. So the post-editor must take out anything which stands out or sounds silly, and must tidy up the word order in places to make the text read more easily, but need not worry if it still sounds a bit awkward and odd in places.

Now in fact this is as far as we got in the one-hour workshop and I attach a list of the guidelines we devised. In future years, however, I would propose to elaborate on this realistic scenario technique so as to carry out carefully defined levels of post-edit on the same text. For example, first would come the raw output as you read a text on the Internet and decided whether it is a suitable one to present to the meeting. Then a rapid post-edit as you show it to a colleague to get their opinion. After that the type of minimal post-edit we have just described, which we called "office quality", suitable for internal discussion. Finally we could imagine that a representative of the business forum wished to report to the Mayor on the matter and a full post-edit would be necessary in order to attach the text to the report. In reality this type of repeated treatment of the same text would involve a lot of wasted effort, but for an exercise it would demonstrate clearly different ways of treating the same MT error.

This describes the activity. To return to the ways in which the MT is assisting language learning, how is this contributing:

- to students' language learning at an appropriate level
- to their translation skills
- to their awareness of communication
- towards giving them a perspective on realistic use of foreign languages at work?

4 Post-editing and language learning

The main objections to introducing MT to language learners are that raw MT output exposes students to incorrect models and does not readily lend itself to correcting those errors. However these objections do not apply to this particular activity as we are working from the foreign language into English, which is the first language of the vast majority of the students. In this way the post-editing activity functions much like classical exercises in reading comprehension and translation. In order to post-edit effectively students must study the text in detail, thereby learning new vocabulary and expressions, new grammatical points and stylistic features. They must acquire a global understanding of the information and function of the text, as they would if they were going to answer questions for a reading comprehension; and they must also grasp the detail of sentence structure and use of vocabulary as precisely as if they were going to translate it from scratch. Far from confusing students or giving them bad habits, the post-editing exercise can actually be seen to have positive benefits for their language learning.

5 Post-editing and translation skills

Here I think the exercise helps in two ways. Firstly, and most obviously, it encourages students to look at any errors and to consider and discuss the best ways of translating them. Some of the incidental discussions, like about the direction of Santa Claus' sleigh, proved useful from the point of view of translation strategy as we talked about

accuracy, fidelity and coherence. Secondly, and perhaps unexpectedly, it can sometimes throw up some good ideas. Of course raw MT output is not normally held up as a model translation, but the fact is that there are some good versions in this example. I would find it hard to find a better version than "*the shop assistant seizes the dollar note with shapeless mittens"* (paragraph 5), which translates the German precisely and vividly. Even "*purchase intoxication"* (paragraph 4), which is a word-for-word translation of *"Kaufrausch",* is worth a second look and bears out what some students have said about using MT in their literary translation courses, namely that they sometimes run a passage through an MT system before starting work. "It gives you ideas," one said; "it gives you some vocabulary," was another comment. At this level students are being encouraged to go beyond the standard dictionary definitions and I think experience improving on an MT translation can give them a bolder, more imaginative outlook and a willingness to consider a non-obvious version.

6 MT and communication skills

Communication skills are amongst the key transferable skills which courses in modern languages aim to develop, and from this point of view consideration of the degrees of post-editing required for different situations is very helpful to enable students to consider the function as well as the content and style of a text. It is surprising how even final-year students can have trouble recognising and articulating the differences between various communication situations and the type of language which is appropriate to each. They will happily define a text type as "literary", "journalistic", or "scientific" but will have much more difficulty defining what they mean by this or characterising the register of a text, and appropriate ways of expressing the same idea in different situations. Often it is only when imagining themselves in a particular situation and comparing the appropriate language that they suddenly become aware of it. Our very detailed scenarios for the business meetings function well in this way and students are more easily able to imagine the type of changes which will be necessary to the text in order to prepare it for a particular readership.

7 MT and languages at work

The distinction between academic and vocational education is a very important one and it is a pity that the debate is often blurred by automatic judgments and strong feelings on one side or the other. On the one hand the proponents of a more theoretical approach to syllabus design minimise the importance of developing practical marketable skills. They equate any attempt to improve students' employability with an attempt to reduce the intellectual rigour of their courses, and reject anything which appears to place constraints on their freedom to decide what they should or should not include. They feel that teachers should train the student to think clearly and to develop a fearless intellectual curiosity, to equip them to be able to apply themselves to anything. Teachers of philosophy or classics tend to this point of view; language courses run along these lines tend to the literary and cultural options, the short, rigorously accurate and creative translations, the tightly argued essay and the structured oral presentation. On the other hand are the teachers of applied disciplines who point out with equal justice that a course of study which has divorced itself from the practice of the subject in the real world is worse than useless to its students. It would be hard to imagine university courses in social work, for example, or agriculture, let alone medicine, which were not informed by the latest practices in these areas and did not give their students practical experience. Language courses of this type will include the more practical tasks like report writing, gist translation and consecutive interpreting skills.

I believe that in the case of machine translation we have an opportunity to create a course which sits very comfortably between these two imperatives, and again the post-editing work we are doing illustrates this. On the one hand the workshop lends itself to discussion of the most abstract principles which underlie the authoring and translating process, by forcing consideration of the human/machine translation divide; on the other hand it gives students their first experience of one of the most practical skills which trainee translators can be expected to acquire.

I am not suggesting of course that our invented scenario actually bears much relation to a real post-editing task. In fact I have not attempted to align our course content too closely to a real-world

translation environmen, for two reasons. Firstly it is very hard to fit in anything like a real post-editing task into the very small number of hours available to us. It is not possible to reproduce in the classroom the kind of large-scale, often repetitive work which would give students a real idea of what it is really like. Secondly, when we are only giving an introduction to the subject, it may be confusing and misleading to look too closely at how these tasks are actually sometimes carried out. I would have liked to give our students some exposure to real post-editing work, but after a visit to a local translation company I decided against it. The work being done involved post-editing of technical documentation produced by a team of translators with the help of a translation memory system. However the post-editing environment was far from the carefully constructed scenario with which we had been working. The company was simply struggling with too many practical difficulties. The communication between members of the team was poor, the company was understaffed relative to the volume of work, and the translation memory database was not maintained, which meant that the individual post-editor was faced with impossibly conflicting guidelines as to how to proceed on any particular case. It would hardly have been fair to expect my students, still familiarising themselves with rules to be applied in clear and ideal circumstances, to make any sense of this type of situation in the time available. After all we are teaching, not training. I would like to hope, however, that our course, balanced between the ideal of theory and the realities of practice, would enable students to recognise and deal with these difficulties as soon as they find themselves in their first job.

Bibliographical References:

Allen, Jeffrey. 2003. Post-editing. Chapter in H. Somers (ed) *Computers and Translation: a Translator's Guide*, John Benjamins, Amsterdam, Netherlands.

Belam, Judith. 2002. Teaching MT evaluation by assessed project work. 6th EAMT workshop, *Teaching Machine Translation.* EAMT/BCS, Manchester, UK

Krings, Hans, translated and edited by Koby, G.S. 2001. *Repairing texts: empirical investigations of machine translation post-editing processes.*Kent State UP, Ohio, USA

Lewis, Derek. 1997. Machine Translation in a Modern Languages Curriculum. *Computer Assisted Language Learning* 10. Intellect, Oxford, UK.

O'Brien, Sharon. 2002. Teaching Post-editing, a proposal for course content. 6th EAMT workshop, *Teaching Machine Translation.* EAMT/BCS, Manchester, UK

Somers, Harold. 2001. Three Perspectives on MT in the classroom. *Workshop on Teaching Machine Translation*, MT Summit XIII, IAMT/EAMT, Santiago de Compostela, Spain

Appendix 1: Texts

German original

Christmas Shopping in NYC

(1) Lange trug New York Trauer – In der Vorweihnacht sind Glanz, Stimmung und Jingle-Bells-Atmosphäre zurückgekehrt. Die Szene mit Shops, Restaurants und Bars hat sich allerdings mal wieder gründlich verändert

(2) Der Weihnachtsmann hebt ab, das der rote Mantel wie ein Drache flattert. Und das immer wieder. Wie von himmlischen Kräften getragen, fliegt der Rentierschlitten durchs Schaufenster, von unten rechts nach oben links, den Goldenen Sternen entgegen. Leise rieselt der Kunstschnee.

(3) Trauben von happy Shopping-Touristen vor dem Kaufhaus stimmen mit den Füßen ab, bleiben ohne Ende staunend stehen.
Ja, so lieben sie New York in der Vorweihnacht. Die Mega-City zwischen Hudson und East River funkelt wieder wie vor den Anschlägen, die sie tief ins Herz trafen. In diesen Tagen ist selbst die stotternde Konjunktur keine entscheidende Bremse mehr.

(4) Die New Yorker müssen diesen Spruch „Shop till you drop", „Einkaufen bis zum Umfallen", erfunden haben. Die Geschäfte sind voll, der Kaufrausch ist zurückgekehrt, unter den diamantenen Lichtern im Artdéco-Diadem des Chrysler Buildings.

(5) Es ist kalt in der Stadt, so richtig kalt geworden. Die Verkäuferin, die mir, dick vermummt, die heibe Brezel vom Rollwagen verkauft, greift mit unförmigen Faustlingen die Dollar-Note. Ein Pferd, das im Strom der Autos eine Kutsche zieht, schnaubt kleine weiße Wolken. Am Rockefeller-Center wärmt gleißen Pracht zumindest die Herzen. In diesem Epizentrum des vorweihnachtlichen New York war Ende November mit einem Sattelschlepper die in Kanada gefällte Riesentanne angeliefert worden. Als der Bürgermeister den Stecker in die Dose steckte, leuchteten 26 000 Lichter am groβten Weihnachtsbaum der Welt. Das gleiche Prozedere wie in jedem Jahr.

(6) New York zum Schauen, Erleben, ja naturlich vor allem Shoppen. Wir setzen Bloomingdale's, Saks Fifth Avenue, Macy's als bekannt voraus und geben Ihnen neue Empfehlungen, auch für den Genuss und günstige Hotels, um den Geldbeutel zu schonen. Kurzfristig gibt es drei und viertägige New York-Angebote der Airlines und Veranstalter bereits um die 600 Euro mit Flug und Hotel.

(7) Bei der Verführung zum Kauf in Plüsch und Pomp wurde an alles gedacht. Auf den Straβen werden Choräle von Met-Mitgliedern gesungen (die Geschäftsleute zahlen), Kaufhäuser in der Fifth Avenue sind mit roten Schleifen zu riesigen Geschenkpaketen verschnürt, und himmlische Einheiten von goldenen Engeln aus Golddraht und Lichterketten komplettieren das Bild. Spaβ am Shopping ist nirgendwo so festlich und verrückt wie auf New Yorks goldener Meile.

MT raw output (Systran Professional Standard)

Christmas Shopping in NYC

(1) For a long time New York mourning carried- gloss, tendency and Jingle Bells atmosphere in the Vorweihnacht returned. The scene with Shops, restaurants and Bars changed however times again thoroughly.

(2) Santa Claus takes off, which the red coat flutters like a Drache. And that again and again. As carried by himmlischen forces, the annuitant carriage flies by the shop window, from down right upward left, the golden stars against. Quietly the art snow rieselt.

(3) Grapes/clusters of happy Shopping tourists before the department store co-ordinate with the feet, stop without end being astonished. Yes, then they love New York in the Vorweihnacht. The megatown center between Hudson and East River sparkles again as before the notices, which met it deeply in the heart. In these days even the stotternde economic situation is no more crucial brake.

(4) The New Yorker ones must have invented this saying "Shop till you drop", "buying up to falling down". The business is full, the purchase intoxication returned, under the diamond lights in the Artdéco artdéco-Diadem of the Chrysler Buildings.

(5) It became coldly in the city, so correctly cold. The shop assistant, who vermummt me, thickly, the hot brezel sold by the truck, seizes the dollar note with shapeless mittens. A horse, which pulls a kutsche in the river of the cars, schnaubt small white clouds. At the skirt skin he center track-sends splendour warms at least the hearts. In this epizentrum at the end of had been delivered to vorweihnachtlichen New York November with a semi-trailer the giant fir gefaellte in Canada. When the mayor put the plug into the box, 26,000 lights at the largest Christmas tree of the world shone. The same Prozedere as in each year.

(6) New York for looking, experiencing, naturally above all Shoppen. We expect Bloomingdale's, Saks Fifth Avenue, Macy's to be familiar and give you new recommendations, also for the benefit and favorable hotels, in order to preserve the purse. There are already at short notice three and four-day new York offers of the airlines and organizers around the 600 euro with flight and hotel.

(7) With the seduction to the purchase in pluesch and Pomp of everything one thought.
Sung on the roads Choraele of Met members (the businessmen to pay), department stores in the Fifth Avenue are tied with red loops to enormous gift packages, and himmlische units of golden angels from gold lead and candlelight demonstrations complete the picture. Fun at the Shopping is nowhere as festive and moved as on new Yorks of golden mile.

English translation

(1) New York wore mourning for a long time. – But in the run-up to Christmas the sparkle, the high spirits and the Jingle-bells atmosphere have returned. The scene, with shops, restaurants and bars, has changed completely once again.

(2) Santa Claus takes off, his red coat fluttering like a dragon's wing. Over and over again. As if pulled by heavenly powers, the reindeer sleigh flies up across the shop window from right to left against a background of golden stars. The artificial snow drifts gently down.

(3) Tight groups of happy tourist shoppers walk around together or stand and stare in endless amazement. Yes, that's how they love New York in the Christmas season. The great metropolis between the Hudson and the East River is glittering again, just like before the events of September 11th, which struck to the very heart of the city. These days even the shaky economic situation is not significantly slowing consumption.

(4) New Yorkers must have invented the phrase "Shop until you drop". The shops are full and shopping fever is back under the diamond lights of the art-deco diadem of the Chrysler Building.

(5) It has got cold in the city, really cold. The salesgirl who sold me a pretzel from a street stand was well wrapped up and seized the dollar note with shapeless mittens. A horse, pulling a carriage through the stream of traffic, breathes out little white clouds. At the Rockefeller Center the glittering splendour is at least heartwarming. Here at the end of November, at the epicentre of New York's Christmas celebrations, an articulated lorry delivered the giant fir tree, felled in Canada. When the Mayor put the plug in the socket, 26 000 lights lit up on the biggest Christmas tree in the world. Just the same as any other year.

(6) There is plenty to look at in New York, plenty to experience, but above all it is for shopping. We take it you know Bloomingdale's, Saks Fifth Avenue, Macy's, and we have new recommendations to make, places to eat, comfortable hotels, to protect your wallet. At the moment airlines and tour operators are offering three- to four-day excursions to New York with flight and hotel for around 600 euros.

(7) In the midst of luxury and splendour, they have thought of everything to tempt you. On the streets the choir of the Metropolitan Opera are singing (the retailers pay the fees). The departments stores of Fifth Avenue are tied into giant parcels with red bows, and groups of golden angels made of gold wire and strings of lights complete the picture. Only the wild festivities of New York's golden mile can make your shopping trip this much fun.

Appendix 2: Post-editing guidelines

For a rapid post-edit, in general: correct only blatant and significant errors.

Do:

- correct any word which has not been translated and does not resemble its TL equivalent. Thus: *himmlischen* > *divine* or *heavenly*
- correct any word which has been translated so wrongly as to make the sentence meaningless. Thus: *annuitant carriage* > *reindeer sleigh*
- delete anything which is confusing, which may take time to correct, and which does not add or detract significantly from the meaning. Thus *Buying up to falling down* is deleted.

Don't:

- change the word order if the meaning is not affected. Thus *the golden stars against* does not change to: *against the golden stars.*
- Change a word which has been wrongly translated, but which still has a related meaning. Thus *the purchase intoxication returned* does not change to *shopping fever returned.*
- Tidy up the style. Thus *And that again and again* does not change to *over and over again.*

For a minimal post-edit, in general: correct anything which stands out, or is disconcerting.

Do:

- change a startling choice to something more bland or appropriate. Thus *seduction* > *temptation*
- Change the word order to make the text flow more easily. Thus *the golden stars against* > *against the golden stars.*
- Change sentence structure where necessary to increase ease of comprehension. For example, split long sentences into shorter ones, or change a noun construction to a verbal one: *the economic situation is no more crucial brake* > *the economic situation is not significantly slowing consumption.*

Don't:

- spend a long time deciding on the very best word or phrase.
- attempt to make the translation sound like a piece of original writing.

A 45-hour Computers in Translation course

Mikel L. Forcada
Departament de Llenguatges i Sistemes Informtics
Universitat d'Alacant, E-03071 Alacant, Spain.
mlf@ua.es

Abstract

This paper describes how a 45-hour Computers in Translation course is actually taught to 3rd-year translation students at the University of Alacant; the course described started in year 1995–1996 and has undergone substantial redesign until its present form. It is hoped that this description may be of use to instructors who are forced to teach a similar subject in such as small slot of time and need some design guidelines.

1 The subject

The subject Computers in Translation (officially *Informática Aplicada a la Traducción*) is a mandatory subject in all official 4-year or 5-year translation degrees in Spain, and can be taught as part of any of the last two years, with a minimum of 4.5 credits (in Spain, a credit is equivalent to 10 h of classroom or laboratory time but can be reduced in some circumstances). The official Ministry of Education descriptor for the subject is very short:

> Access to the necessary tools for translation work. Machine translation and computer-assisted translation. System integration.

This leaves a lot of freedom to design the actual syllabus. Some universities extend these 4.5 credits and split the subject in two or more courses; other universities add mandatory "computer literacy" courses during the first two years. The University of Alacant simply satisfies the minimum requirements: a single 45-h Computers in Translation course is programmed as part of the 3rd year; this course is expected to provide future translators with enough knowledge and skills about the application of computers to translation.

2 Students, groups and sessions

In Alacant, we have about 150 students each year from German (30), English (60) and French (60) translation studies, who meet in either of two 75-student classroom groups and in one of six 25-student laboratory groups, regardless of the language. The course is organized in 30 1.5-hour sessions (15 weeks, 2 sessions a week): 19 sessions are held in the classroom and 11 in a computer laboratory. In addition, each instructor is in his or her office during a publicly-announced schedule of six hours a week for students to ask questions and clear doubts, either by visiting personally or through the University's *virtual campus* facilities.

3 Methodology in brief

3.1 Classroom work

Classroom work is organized around an *activity program*, a sequence of *activities* designed for students to learn the key concepts and basic techniques of the subject. Activities basically pose open problems that have to be tackled by the students, before the *theory* is explained. Here's an example of an introductory activity:

> Ambiguity is an essential feature of natural languages. Could you write up a formal definition of ambiguity? Why do you think human language is ambiguous? Why does ambiguity make machine translation difficult?

which is followed by a more detailed activity in which students are given a set of carefully chosen

ambiguous sentences (with slight hints to clarify the various interpretations) and are asked to use those sentences and other examples they may come up with to design a linguistically motivated classification of ambiguity types.

Students organize themselves in groups of three (stable, if possible, for the duration of the course) to perform the activities; groups designate a spokesperson for each session. After working individually (sometimes at home, before the session) and in groups, a discussion takes place in the classroom, followed by a "classical" lecture segment where classroom work is integrated with the explanation given by the instructor. For example, after the ambiguity activities described above, the instructor introduces the *principle of semantic compositionality* (Radford et al. 1999) and therefore classifies ambiguities as *lexical* (the sentence has more than one interpretation because one or more words do: *polysemy*, *homography*, *anaphors*), *structural* (the sentence has more than one interpretation because it has more than one possible parse tree: *adjunction* and *coordination* ambiguities, ambiguity due to *Wh-movement*), or *mixed* (when both things happen simultaneously).

This way of organizing classroom work (based on a proposal used to teach natural sciences, Gil-Pérez and Carrascosa-Alis 1994) gives students the opportunity to analyse each problem and even advance parts of the solution, and prepares them to receive the solution when the teacher explains it after the discussion. But it also gives the instructor very valuable information on what students already know which he or she can use to *anchor* (Clement et al. 1989) the explanation of new, sometimes rather complex, concepts.

After the session, sudents are expected to make a synthesis between their individual and work group, the classroom discussion, and the explanations by the instructor, using the recommended literature and the office hours of instructors. The use of office hours to clear as soon as possible any doubts is strongly encouraged by instructors to avoid pre-exam *indigestion*.

3.2 Laboratory work

In each laboratory session, a computer assignment is proposed which has to be performed either individually or in pairs. For example, in unit 7 (lab session L_6) students are asked to analyse what does a given commercial MT system do besides simply substituting words, by first forcing the system to translate in isolation (e.g., each one in a paragraph) the words of a set of sentences and then translating the whole sentences (Pérez-Ortiz and Forcada 2001). In another assignment (lab session L_7), students feed a set of increasingly complex noun phrases designed by the instructors into an MT system to incrementally infer the word-reordering rules used by explaining the resulting correct or incorrect translations (Forcada 2000). The following section gives brief descriptions of the remaining laboratory assignments.

4 Syllabus

The current design of the syllabus started in 1995, before proposals like LETRAC (Badia et al. 1999) or surveys about the teaching of these matters (Balkan et al. 1997) were available. I basically interpreted the official description of the subject and made a quick survey of what other universities in Spain were doing (according to their webpages). After eight years of redesign, the course was eventually structured in 10 units or *blocks* ($\mathbf{B}_1 \ldots \mathbf{B}_{10}$), which will be briefly described and commented upon in this section. Classroom sessions are denoted $C_1 \ldots C_{19}$, lab sessions are denoted $L_1 \ldots L_{11}$; sessions C_{19} and L_{11} are spare sessions for doubt clearing, finishing laboratory assignments, or just to make up for a session which might have been postponed or cancelled (strikes, torrential rain).

4.1 The ten "blocks" in brief

Block: $\mathbf{B}_1$: What are we going to study?

Objective: Knowing the ways in which computers may be applied to translation; recognizing which parts of the translation task can be automated and which ones cannot; understanding the concept of *machine translation*; being able to distinguish the two main kinds

of *computer-assisted translation*: *human-aided machine translation* and *machine-aided human translation*; being able to enumerate and describe computer tools useful for translation.

Classroom sessions: C_1 (week 1).

Lab sessions: None.

Block $\mathbf{B}_2$: Computers and programs.

Objective: Acquiring basic concepts about how personal computers work, to improve their practical application and the understanding of their applications to translation: hardware and software; memory units (b, B, kB, MB, GB); RAM, magnetic and optical media; files, directories and directory structure; computer programs and instructions; CPUs, frequency, speed, and instruction sets; operating systems.

Classroom sessions: $C_2 - C_4$ (weeks 2 and 3).

Lab sessions: L_1 (week 3: analysing the hardware characteristics of the PC in the lab; creating and modifying a directory structure on a diskette).

Block $\mathbf{B}_3$: Internet basics.

Objective: Acquiring basic concepts about the internet and about its application to the translation task: computer networks, internet as a network of networks, internet services of interest to translators (lexical databases, dictionaries, encyclopedia, texts, bitexts), URLs, IPs, names and domains, hardware and software needed for home and office access to the internet, the use of search engines to choose among various possible translations.

Classroom sessions: C_5 (week 3).

Lab sessions: L_2 and L_3 (weeks 4 and 5: searching for translations with Google; basics of HTML; building a webpage from a template and publishing it).

Block $\mathbf{B}_4$: Texts and formats

Objective: Learn basic concepts about the storage, format, structuring, presentation, creation and manipulation of text documents: character encoding (ANSI, the ISO-8859 family, Windows CP-1252, Unicode, UTF-8), the use of format for presentation and structuring of content, and conflicts between the two objectives), common formats and their usage (RTF, HTML, PostScript, PDF, etc.); XML (well-formedness, validation using a DTD, separation of content and presentation through stylesheets); generating text through digitization and OCR or through speech recognition.

Classroom sessions: C_6 and C_7 (weeks 4 and 5).

Lab sessions: L_4 and L_5 (weeks 6 and 7: validating XML documents against a simple DTD; tagging a text according to a certain DTD and validating it).

Block $\mathbf{B}_5$: Machine translation and applications

Objective: Learning what machine translation is and how it can be used in the real world despite its imperfections: assimilation and dissemination applications; human-aided machine translation (preediting, postediting, interaction, controlled languages); MT as a a component of communication systems (multilingual chat, translated web browsing); nonlinguistic requirements (speed, format preservation).

Classroom sessions: C_8 (week 6).

Lab sessions: none.

Block $\mathbf{B}_6$: Ambiguity[1]

Objective: Identifying ambiguity as the main source of errors in machine translation, understanding the diversity of its mechanisms and learning to classify the ambiguity of a given sentence or statement: ambiguity through the principle of semantic compositionality; lexical ambiguity (homography, polysemy, anaphor, anaphor through empty categories), structural ambiguity (adjunction, coordination, movement of constituents), and mixed lexical-structural ambiguities; basics of ambiguity resolution in MT systems (statistical and rule-based methods).

Classroom sessions: C_9 and C_{10} (weeks 7 and 8).

Lab sessions: none.

[1]The fact that a complete block is devoted mainly to linguistic aspects may be surprising; however, we have found that our students arrive with severe deficiencies in basic linguistics, even after a mandatory subject called "Linguistics applied to translation". Discussing ambiguity is an excellent way to review basic concepts needed to understand the ensuing blocks.

Block $\mathbf{B}_7$: How does machine translation work?

Objective: Knowing the main machine translation strategies and their implementation as distinct, consecutive phases or tasks; identifying these strategies by analysing the machine translation of real or synthetic texts (using a mechanical word-at-a-time, word-for-word translation called *model zero* as a reference model, Pérez-Ortiz and Forcada 2001): commercial systems as intuitive refinements over *model zero* (categorial homograph resolution, adding multiword expressions to dictionaries, rules for local reordering and agreement); the transfer architecture (analysis, transfer and generation; morphological, syntactic and semantic transfer, intermediate representations, linguistic information needed in each phase, modularity as an advantage); interlingua as null transfer and its advantages; inductive strategies (statistical MT, example-based MT).

Classroom sessions: $C_{11} - C_{14}$ (weeks 9 and 10).

Lab sessions: L_6 and L_7 (weeks 8 and 11: "machine translation is not word by word", Pérez-Ortiz and Forcada 2001, and "discovering reordering and agreement rules", Forcada 2000).[2]

Block $\mathbf{B}_8$: Machine translation evaluation

Objective: Learning to use knowledge about how MT systems work to evaluate them with an adequate technical level and well-founded criteria: aspects to be evaluated and their relative importance (quality, ease of use, extensibility, speed, memory usage), the difficulty of quality evaluation through postediting, the inadequacy of comparison with human translation, *predictive evaluation* after careful error diagnosis.

Classroom sessions: C_{15} (week 11).

Lab sessions: L_8 (week 12: evaluation and classification of MT errors in real texts).

Block $\mathbf{B}_9$: Lexical databases

Objective: Learning basic concepts about databases and applying them to lexical or terminological databases: tables, records, fields, index fields, ordering for faster search (dichotomic search), indexing for multiple orderings, updating; using lexical databases for specialized translation and terminological coherence; concept-based lexical databases and their fields (terms, definitions, subject, author, date, cross-references). Being able to design, create and maintain a lexical database using the suitable software.

Classroom sessions: C_{16} (week 12).

Lab sessions: L_9 (week 13: creating a small lexical database and performing searches over it).

Block $\mathbf{B}_{10}$: Translation memories

Objective: Understanding the importance of translation memories (TM) as an efficient solution to human translation with a high degree of repetitiveness: TMs as databases (translation units as records); bitext processing (segmentation rules, semiautomatic alignment, translation unit extraction); pre-translation (exact and approximate matches), advantages of TM-based translation work; the TMX standard.

Classroom sessions: $C_{17} - C_{18}$ (weeks 13 and 14).

Lab sessions: L_{10} (week 14: a taste of the complete TM cycle: alignment of a bitext followed by pre-translation and correction of a new text and TM updating).

4.2 Comparison to LETRAC

It is inevitable to compare, even if briefly this syllabus to the only detailed curriculum proposal available, LETRAC[3] (Badia et al. 1999):

- Forty-five hours forces us to make sacrifices and makes it impossible to include a great part of the LETRAC proposal. One could say that the weight given to Computers in Translation in Alacant does not match the importance given by LETRAC to the subject.

[2] The first assignment is programmed before actual work starts in the classroom and serves as a very nice introduction to the first activities of $\mathbf{B}_7$.

[3] It is surprising to see that translator associations like the International Translators Federation (FIT-ITF) and the American Translator Association (ATA) have no curriculum proposals of their own. There is another initiative (`http://www.lisa.org/leit/`) by LISA (Localization Industries Standards Association) but it is centered around *localization* and has therefore a narrower scope than the subject discussed here.

- The official description of Computers in Translation makes the study of MT mandatory, whereas LETRAC makes it optional (LETRAC, for example, gives more weight to translation memory).
- Desktop publishing (QuarkXPress, Framemaker, Pagemaker, Ventura, etc.), mandatory in LETRAC, is not taught in Alacant.
- LETRAC barely touches XML (maybe it was too early: it does mention SGML) and gives more weight to character encoding than to structure and presentation (in Alacant they have similar weight).
- The treatment of terminology is wide in LETRAC and very brief in the Alacant subject, which may be compensated for by mandatory subjects dealing with terminology totalling 10,5 credits.

5 Bibliography

Initially the course relied on the classical book by Hutchins and Somers (1992), articles such as (Hovy 1993), and a set of handouts which have eventually grown into a downloadable *textbook* or *lecture notes* (`http://www.dlsi.ua.es/~mlf/iat/iat.pdf`) in Catalan.[4] Students are strongly encouraged by instructors to read other textbooks and materials (just to cite a few: Arnold 1993; Boitet 1996a; Boitet 1996b; Hutchins 2001; Hutchins 1996; Jacqmin 1993; Krauwer 1993; Lewis 1997; Nirenburg 1987; Sager 1993; Samuelson-Brown 1996; Somers and Rutzler 1996; Trujillo 1999; Vandooren 1993; Wojcik and Hoard 1996) in addition to class notes as a way of acquiring alternate views about the subjects and testing their comprehension of basic concepts.

6 Closing comments

It is hard to describe a whole course in half a dozen pages; this summary is presented in the hope that it may be helpful to instructors teaching in similar environments or facing similar time restrictions; in fact, all of the materials (in Catalan) are available to anyone interested (I could even consider translating selected materials into English on request).

[4] Catalan and Spanish are the official languages of the University of Alacant; the course is taught in Catalan.

Acknowledgements: I thank Andy Way for comments and suggestions on the manuscript.

References

Arnold, D. (1993). Sur la conception du transfert. In Bouillon, P. and Clas, A., editors, *La traductique*, pages 64–76. Presses Univ. Montréal, Montréal. English translation available: `http://clwww.essex.ac.uk/~doug/papers/transfer.ps.gz`.

Badia, T., Freigang, K.-H., Haller, J., Horschmann, C., Huber, D., Maia, B., Reuther, U., and Schmidt, P. (1999). LETRAC curriculum modules. Available at `http://www.iai.uni-sb.de/letrac/home.html`.

Balkan, L., Arnold, D., and Sadler, L. (1997). Tools and techniques for machine translation teaching: A survey. Technical report, University of Essex. Available at `http://clwww.essex.ac.uk/group/projects/MTforTeaching/`.

Boitet, C. (1996a). (human-aided) machine translation: a better future? In Cole, R., editor, *Survey of the State of the Art in Human Language Technology*. Available at `http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html`.

Boitet, C. (1996b). Machine-aided human translation. In Cole, R., editor, *Survey of the State of the Art in Human Language Technology*. Available at `http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html`.

Clement, J., Brown, D., and Zietsman, A. (1989). Not all preconceptions are misconceptions: finding 'anchoring conceptions' for grounding instructions on students' intuitions. *International Journal of Science Education*, 11:554–565.

Forcada, M. (2000). Learning machine translation strategies using commercial systems: discovering word-reordering rules. In *MT 2000: Machine Translation and Multilingual Applications in the New Millennium*, pages 7.1–7.8, Exeter, UK.

Gil-Pérez, D. and Carrascosa-Alis, J. (1994). Bringing pupils' learning closer to a scientific construction of knowledge: A permanent feature in innovations in science teaching. *Science Education*, 78:301–315.

Hovy, E. (1993). How MT works. *Byte*, (enero):167–176.

Hutchins, J. (1996). Evaluation of machine translation and translation tools. In Cole, R., editor, *Survey of the State of the Art in Human Language Technology*. Available at `http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html`.

Hutchins, J. (2001). Machine translation over fifty years. *Histoire Epistémologie Langage*, 23(1):7–31.

Hutchins, W. and Somers, H. (1992). *An introduction to machine translation*. Academic Press. (Spanish translation available, published by Visor in 1995).

Jacqmin, L. (1993). Classification générale des systèmes de traduction automatique. In Bouillon, P. and Clas, A., editors, *La traductique*. Presses Univ. Montréal, Montréal.

Krauwer, S. (1993). Evaluation of MT systems: a programmatic view. *Machine Translation*, 8.

Lewis, D. (1997). MT evaluation: science or art? *Machine Translation Review*, 6:25–36.

Nirenburg, S. (1987). *Machine translation: Theoretical and methodological issues*. Cambridge University Press, Cambridge.

Pérez-Ortiz, J. A. and Forcada, M. L. (2001). Discovering machine translation strategies beyond word-for-word translation: a laboratory assignment. In Forcada, M., Pérez-Ortiz, J., and Lewis, D. R., editors, *MT Summit VIII, Proceedings of the Workshop on Teaching Machine Translation*.

Radford, A., Atkinson, M., Britain, D., Clahsen, H., and Spencer, A. (1999). *Linguistics: an introduction*. Cambridge Univ. Press, Cambridge.

Sager, J. C. (1993). *Language engineering and translation: consequences of automation*. Benjamins, Amsterdam.

Samuelson-Brown, G. (1996). New technology for translators. In Owens, R., editor, *The translator's handbook*. Aslib, London, 3rd edition.

Somers, H. and Rutzler, C. (1996). Machine translation. In Owens, R., editor, *The translator's handbook*. Aslib, London, 3rd edition.

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

Vandooren, F. (1993). Divergences de traduction et architectures de transfert. In Bouillon, P. and Clas, A., editors, *La traductique*. Presses Univ. Montréal, Montréal.

Wojcik, R. and Hoard, J. (1996). Controlled languages in industry. In Cole, R., editor, *Survey of the State of the Art in Human Language Technology*. Available at `http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html`.

Teaching Statistical Machine Translation

Kevin Knight

USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey CA 90292
knight@isi.edu

Abstract

This paper describes some resources for introducing concepts of statistical machine translation. Students using these resources are not required to have any particular background in computational linguistics or mathematics.

1. Introduction

This paper describes three resources for introducing concepts of statistical machine translation. The first consists of parallel corpora and tools to support a human translation task. The second is a tutorial workbook. The third is a multiple translation corpus put out by the Linguistic Data Consortium. Working with these resources requires no particular background in computational linguistics or mathematics.

2. Human Translation Using Parallel Corpora

Computers look at parallel corpora very differently from people. To help get students into the "mind" of an automatic statistical translator, we put together a small bilingual corpus (12 sentence pairs) where the language pair is "Centauri/Arcturan" ([Knight, 1997] contains the corpus). None of the words in this corpus are understandable to humans.

Furthermore, students are given three new Centauri sentences to translate. Their only resource is the bilingual text. This problem can be solved in a few hours. Students learn:

- What parallel corpora look like.
- To view parallel corpora through the eyes of a computer.
- How parallel corpora are relevant to machine translation.
- How to build bilingual dictionaries from parallel corpora.
- How cognate information may be useful in machine translation.
- How to do word alignment, and how to employ the pigeonhole principle.
- About the chicken-and-egg nature of dictionaries (which enable word alignments) and word alignments (which enable dictionary building).

These concepts can be learned without any prior instruction – students have to learn them to solve the task. Later, it can be revealed that the "Centauri/Arcturan" corpus is really a lightly disguised Spanish/English corpus.

A somewhat larger example of the same exercise can be found at:

www.isi.edu/natural-language/mt/contest

This houses a collection of 1100 real English/Tetun sentence pairs (Tetun is a major language of East Timor), plus a monolingual Tetun news article of 10 sentences, to be translated/decoded by hand. A search tool is provided

that returns all sentence pairs containing any requested monolingual word or phrase. Students learn:

- About word alignment and dictionary building at a larger scale.
- About phrase-to-phrase alignment, the norm in real translation data.
- About unalignable function words.
- The importance of knowing the target language (versus source) in making fluent translations.
- The importance of short sentence pairs (where alignment possibilities are restricted) in helping disambiguate/align longer sentence pairs.
- About locality in word order shifts.
- How to guess the meanings/translations of unknown words.
- About how much uncertainty the machine faces in working with limited data.

Tetun has non-standardized spelling (the name East Timor is spelled seven different ways in this Tetun corpus) but virtually no morphological inflection.

Students may also get ideas about machine translation algorithms after doing the job manually. [Al-Onaizan et al, 2000] describes a translation contest using this corpus, and gives results of de-briefing the winners.

3. Statistical Machine Translation Tutorial Workbook

This short workbook (www.isi.edu/~knight) gives a gentle introduction to the IBM statistical MT models, presenting the concepts in [Brown et al 93]. The presentation is mathematically from scratch, and many exercises are provided. Students learn:

- About Bayes Rule and noisy-channel probabilistic models.
- About n-gram language models and smoothing.
- About generative translation modeling, in particular Model 3 from [Brown et al 1993].
- About automatic hidden parameter estimation and word alignment via the EM algorithm.

This workbook was written in 1999 and already lacked in-depth discussion of decoding algorithms. Some significant advances in statistical MT since then include phrasal translation models, maximum entropy models, and alternatives to training using maximum likelihood. However, completing this workbook is good background for understanding this subsequent work.

4. Multiple Translation Corpus

The Linguistic Data Consortium (www.ldc.upenn.edu) has issued a very interesting data set of 100 Chinese news texts with 10 English translations each. It is called the Multiple Translation Corpus (MTC). This data has been used for automatic MT evaluation, following [Papineni et al, 2002], and also for paraphrasing research [Pang et al, 2003]. It is interesting in its own right, and by studying it, students learn:

- There is no right answer in translation, but there are wrong answers. Translators make a lot of mistakes.
- There is a lot of variation in translation, but by the ninth translation, a surprisingly large amount of that variation has already been observed.
- There are many phrases that do not admit variation, and all translators use the same wording.

This corpus is also a good jumping-off point for the algorithmic/statistical study of automatic MT evaluation.

5. References

[Brown et al, 1993] "The Mathematics of Statistical Machine Translation: Parameter Estimation", P. Brown, S. Della Pietra, V. Della Pietra, R. Mercer. *Computational Linguistics*, 19(2).

[Knight, 1997] "Automating Knowledge Acquisition for Machine Translation", K. Knight, *AI Magazine*, 18(4).

[Al-Onaizan et al, 2000] "Translating with Scarce Resources", Y. Al-Onaizan, U. Germann, U. Hermjakob, K. Knight, P. Koehn, D. Marcu, K. Yamada, AAAI-2000.

[Pang et al, 2003] "Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences," B. Pang, K. Knight, and D. Marcu. NAACL-HLT-2003.

[Papineni et al, 2002] "Corpus-based Comprehensive and Disagnostic MT Evaluation: Initial Arabic, Chinese, French, and Spanish Results", K. Papineni, S. Roukos, T. Ward, J. Henderson, F. Reeder. NAACL-HLT-2002.

Teaching Machine Translation in a Graduate Language Technologies Program

Teruko Mitamura, Eric Nyberg, Robert Frederking
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{teruko, ehn, ref}@cs.cmu.edu

Abstract

This paper describes a graduate-level machine translation (MT) course taught at the Language Technologies Institute at Carnegie Mellon University. Most of the students in the course have a background in computer science. We discuss what we teach (the course syllabus), and how we teach it (lectures, homeworks, and projects). The course has evolved steadily over the past several years to incorporate refinements in the set of course topics, how they are taught, and how students "learn by doing". The course syllabus has also evolved in response to changes in the field of MT and the role that MT plays in various social contexts.

1 Introduction

The Language Technologies Institute (LTI) is a unit within the School of Computer Science at Carnegie Mellon University. The LTI has offered MS and Ph.D. degrees in language technology since 1996[1]. The core LTI curriculum consists of four focus areas (Linguistics, Computer Science, Statistical/Learning and Task Orientation); Ph.D. students are required to take at least one course from each of the four focus areas. The Task Orientation focus, which also includes Information Retrieval, Speech Recognition, and Software Engineering, has always included a Machine Translation course. The MT class typically has an enrollment of about 15 students. The curriculum also contains an MT Lab course, where students complete hands-on exercises related to the MT course lecture material (see Section 5).

The role of MT in teaching varies, depending on the nature of the students and the goals of the instruction (Somers, 2001). The students may include translators, second language learners, or university students with different academic orientations (computer science, computational linguistics, etc.). The goals of instruction might focus on the use of existing MT tools in particular applications (e.g., teaching job skills to translators), the use of MT in second-language learning, or how to create new MT systems that surpass existing approaches.

The students who take our course are generally M.S. or Ph.D. students in the Language Technologies Institute with a background in computer science or linguistics. All students are expected to write programs while taking courses at LTI, and students who lack a programming background are expected to take appropriate preparatory courses. The goals of the MT course are primarily technical, and focus on teaching students how to develop new MT systems. We also attempt to infuse the students with an appreciation of the business issues surrounding successful deployment of MT systems. We do not, however, spend much time teaching specific MT tools or commercial software to the students (although students may elect to evaluate a commercial tool as part of a term project).

In the remainder of the paper, we describe the evolution and current status of the course, and present some of the challenges we have encountered while teaching the course.

[1]The LTI was created as an expansion of the earlier Center for Machine Translation (CMT), which existed at CMU from 1986 to 1996.

2 Objectives

There are two sets of objectives for the course: specific objectives related to the MT subject area, and general objectives associated with the graduate programs. The main objectives within the MT subject area include:

- Obtain a basic understanding of MT systems and MT-related issues;
- Learn about the theory of MT and approaches to MT;
- Learn about basic techniques for MT development, in preparation for the MT Lab course and real-world MT system project development;
- Obtain in-depth knowledge of one current topic in MT, or perform an analysis of a given MT problem, matching it with the most suitable techniques.

The general objectives include learning how to find an interesting research topic, learning to conduct a research investigation, and learning to organize and present research results. Students are required to give final presentations to the entire class, and in-class discussion of individual research projects is an important educational experience in the course. Through their term project work, students begin to learn how to become MT independent researchers.

3 What We Teach

There are no formal prerequisites for the MT course, but students are encouraged to take two other LTI courses ("Grammars and Lexicons", "Algorithms for NLP") before they enroll in the MT course. The course reading materials consist of a compilation of articles, chapters and papers from various textbooks, technical reports, and published papers, along with pointers to relevant web sites and conference proceedings. Basic reading on MT is taken from various MT-related texts, such as Hutchins and Somers (1992), Arnold et. al. (1994), and Trujillo (1999). Readings on MT history are selected from Hutchins (1986) and Hutchins (2000). A chapter from Manning and Schütze (1999) is used for Statistical MT. Goodman and Nirenburg (1991) and Dorr (1993) are used for background on linguistic knowledge development. We also provide relevant technical papers from journals and conferences to augment the lectures described below. The course is divided into the following sections:

Introduction to MT: This section provides basic background on MT, including history, fundamental approaches, and examples of MT systems.

Modern Theory and Approaches for MT: This section includes separate lectures on various approaches to MT, including Direct and Transfer Methods, Interlingual MT, Example-Based MT, Statistical MT and Multi-Engine MT. Technical papers provided to the students include Mitamura et. al. (1993), Carbonell et. al. (1992), Brown, R. (1996), Collins et. al. (1996), Brown, P. et. al. (1990), Berger et.al. (1996), Frederking et. al. (2000), and Brown and Frederking (1995).

MT System Development: This section provides a software engineering perspective on MT development, with an in-depth examination of how to build a working MT system. We present various software engineering activities (e.g., Domain Analysis, Requirements Specification, MT Code Development, MT Knowledge Development), primarily illustrated with examples from rule-based, interlingual MT applications. We also discuss some of the modular software architectures that are used for analysis and generation (e.g. Reiter and Dale, 1995). For MT knowledge development, we use the KANT system (Mitamura et. al, 1993) to introduce a grammar formalism and transfer rules for analysis and generation. This section includes a homework assignment where each student builds a small MT system that translates from English to a language that he or she is familiar with.

Topics in MT: After presenting the basics of MT systems, the course covers additional topics in more detail:

- **Ambiguity and Ambiguity Resolution:** Various ambiguity issues are discussed, including lexical, structural and semantic ambiguities which cause problems for MT. Various ambiguity resolution methods (e.g., word sense dis-

ambiguation, structural disambiguation) are introduced. Different aspects of ambiguity resolution in the KANT system are covered in a case study (Mitamura, et. al. 1999; Baker, et. al. 1994).

- **Controlled Language (CL) Input/Output:** This lecture covers the definition of CL, the goals of CL, and different types of CLs. The history of CL is presented, followed by a discussion of the issues in design, development, and deployment of CL vocabulary and grammar for MT. We also discuss the success criteria for deploying CL in a real-world context (Nyberg, et al. 2003).

- **MT Evaluation:** This lecture introduces commercial goals and research goals in MT evaluation, placed in the context of the history of MT evaluation. Traditional evaluation approaches (e.g. the DARPA MT evaluations in the '90s) and more recent evaluation measures (IBM's BLEU Metric; Papineni, et. al. 2002) are presented. We also discuss the relationship between MT quality and associated postediting costs, and point out that evaluation in industry is often based purely on cost savings rather than an independent quality measure. The other factors related to MT system selection for a specific business solution are addressed in a separate lecture on the Business Case for MT (see below).

- **MT Workflow and Human Factors:** In this lecture we discuss the relationship between an MT system and its context of use, whether by an organization (workflow), by an individual (human factors), or within other software systems (embedded MT systems). The human factors lecture uses two optional readings, a standard human factors reference (Nielsen 1993) and the online Apple Human Interface Guidelines text[2].

- **Business Case for MT:** In this lecture we focus on how to judge the utility of MT systems from a business perspective. Depending on the characteristics of the problem to be solved (language pair(s), time to market, domain, translation volume, hardware/software integration, etc.) what is considered an effective MT solution may vary widely from customer to customer. Students learn how to calculate return on investment (ROI) based on business parameters extracted from customer interviews and estimates of how MT will speed up the translation process.

- **Commercial MT Systems:** This lecture presents commercial MT systems. We begin by discussing customer requirements issues. We then describe the range of MT development options, from fully custom systems, through customizable systems, to straight off-the-shelf systems with no customization. We describe six specific systems in some detail, covering a variety in terms of complexity and age.

- **Speech-to-Speech MT System Development:** We base this lecture largely on local research work in Speech-to-Speech MT, both the interlingua-based (Levin et al, 1998; Lavie 1996; Levin et al, 2000; Lavie et al, 2002) and EBMT-based (Frederking et al, 2000; Frederking et al, 2002) lines of research. We describe the effects of spoken input and speech recognition technology on the design, use, and evaluation of MT systems.

Term Project Presentation and Discussion: The last section of the class is devoted to student presentations of their term projects and group discussion of each project. The class presentation is made before the final written report is due, so that the students can refine their report based on feedback and ideas from other students and the course faculty.

4 How We Teach

The current MT course combines a series of lecture presentations (see topics in previous section) with homework assignments, examinations, and a multi-faceted term project. These activities are discussed in more detail in this section.

The homework assignments typically include a mixture of linguistic analysis (e.g. on lexical mis-

[2] `http://devworld.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-15.html`

matches in a language pair) and hands-on creation and evaluation of MT systems. During 2003, the course included the following homework exercises:

- **Exploring MT systems on the web**. Students are asked to locate a set of MT systems or translation services available on the web. Students must a) categorize them according to important attributes such as type of service, application domain, languages, integration requirements, costing, etc.; b) Perform a comparative evaluation of two MT systems on the same language pair.
- **Building a simple transfer-based MT system**. Using the CMU Generalized LR Parser (Tomita, et al.,1988), Transformation Kit and Genkit (Tomita and Nyberg, 1988), students write a simple transfer-based MT system for two language pairs: English to English paraphrasing, and an additional target language chosen by the student. The same analysis grammar is used for both language pairs. The final deliverable includes an analysis of the system's performance and issues of transfer with the chosen language.

The course also includes two closed-book, in-class examinations (a midterm and a final). The exams contain primarily essay questions, with occasional questions that require simple mathematical calculations (e.g., a return on investment scenario for MT deployment).

The capstone activity in the course is the term project, which includes the following activities:

- **Directed Research**. Students work with their term project advisor to define a set of research activities, discuss possible approaches, and construct a research plan for the semester.
- **Written Report**. Each student must write and submit a final report detailing the research and development (if any) they accomplished during the course. If the project involved empirical analysis (e.g. of MT system performance), the experimental design and results are included. Students may also elect to do a survey of the recent literature in a particular area of MT research (e.g. problems of transfer for a specific language pair).
- **Public Presentation and Discussion**. Students present their project and results in a public presentation at the end of the semester. Each project is discussed by the instructors and other students in a discussion period that follows the presentation. Students receive feedback on their results and research methodology, and gain experience with public speaking.

Examples of recent term projects include: a) "An Evaluation of the BLEU and NIST Metrics for Automatic Evaluation of MT" (comparative analysis of proposed standards); "English-Portuguese Translation Using a Transfer-rule Based System" (a solution for a particular set of transfer problems); "Current Research Projects on Translating to Asian Languages" (a survey of recent work); "Lattice Input for Speech-to-Speech MT" (technical project on extending an existing system); "A Framework for Pay-Per-Translation With On-Board EBMT System" (exploratory prototype).

Although traditional assessments such as homeworks and exams are still a fundamental part of teaching the MT course, we have tried to evolve the course in the direction of "learning by doing". Students gain experience with "real world MT", either by building MT solutions or evaluating existing MT research and development efforts. We feel that hands-on experience is essential if the course is to adequately prepare the student to work with MT technologies after they complete their degree.

5 MT Lab

In order to complement the MT lecture course with hands-on, in-depth exploration of various MT paradigms, we created a companion course, title 'Self-Paced Lab in MT Algorithms'. The lab course has two options or 'tracks' that may be selected by the students:

- *On-line Course Materials and Exercises*. Students may choose to complete a web-based course which includes exercises on parsing, generation, and semantic interpretation in a KBMT framework. Students are given sample code libraries and partially implemented systems to work with, so that the amount of coding required per student is appropriate. Some programming support effort is required to keep the

Track	PROs	CONs
On-Line Web-Based Exercises	• Well-defined, repeatable, course matures over time	• Limited number of topics (1 language pair, 1 MT approach) • Requires programmer support to maintain code libraries
Supervised Independent Study	• Broad range of topics (language pairs, MT approaches, applications) • Flexible Curriculum: Students can define course content that fits their research goals	• Requires time commitment from faculty supervisors

Figure 1: **The PROs and CONs of the two MT Lab tracks.**

code libraries running on platforms accessible to the students.

- *Supervised Independent Study*. With pre-approval from the course instructor, students may also define their own hands-on project, and select a project advisor from among the program faculty. Projects may explore a wide range of MT approaches (Statistical MT, EBMT, Multi-Engine MT, etc.), application areas (speech MT, mobile MT, web MT), and language pairs. Additional support comes from advisor's research team, if required. At the end of the course, students present their project in a presentation/demo for the entire class, which promotes sharing of research results.

When the lab course was first taught in the late 90's, only the on-line web-based track was available. More recently, students have been actively encouraged to propose independent project topics. The result is that the lab course content can be tailored to better meet the research goals of individual students. Students choosing the independent study track work closely with a faculty advisor, typically in the advisor's area of expertise. The incremental cost of adding this track is the time spent by the faculty members who contribute their time to the course as project supervisors. The PROs and CONs of the two tracks are summarized in Figure 1.

In a department with limited resources for teaching MT, it may be more feasible to base a lab course on pre-defined, web-based exercises with supporting code. Even when an independent study option is available, some students may still select the web-based option when they chiefly desire a highly structured, guided course. Nevertheless, it has been our experience that students prefer the more flexible approach of the supervised independent study. The additional motivation students feel for a self-defined project can often lead to a greater degree of accomplishment during the course. The problems and challenges they face in a self-defined project may be more difficult than those present in the pre-defined exercises, but afford a more realistic exposure to the realities of MT research and development.

6 Discussion

In this section, we present some of the issues we have experienced in teaching our MT course. Since most of the students at LTI have a computer science background, our course is designed for students who are able to work with existing tools (e.g. parsers, generators, etc.) effectively after a brief introduction. However, since we only recommend two courses prior to the MT course (see Section 3) and do not require them as absolute prerequisites, we sometimes teach students with a different background (e.g. linguistics) or levels or preparation (e.g. motivated undergraduates). This is a challenge for the instructors, and we sometimes find it necessary to offer special help sessions outside the lectures. Nevertheless, we have found that most students that are well-motivated and apply themselves are able to pass the course without difficulty.

The course lectures include various focused topics in MT; among them, the lecture on "Business

Case for MT" is unusual for our students, because it focuses on business issues rather than technology. Nevertheless, we feel it is important to make students aware of business perspectives on MT as part of a general education on MT as a practical area of application.

The most interesting part of the course is the term project, which covers a wide range of topics due to the varied interests of the individual students. Sometimes, MT is embedded inside a larger system to provide a multilingual capability (e.g. "Keyword Selection for Multi-Lingual Question Answering"). Some of the embedded applications of MT are often novel and sometimes unique; for example, a recent project explored "Intra-Language Matching of Proteins", where MT techniques were adapted to protein analysis. Since more recent applications of MT go far beyond technical document translation, we have adjusted the term project requirements to accept a wide variety of MT-related topics.

To date, the course has not given much emphasis to tools, environments and techniques for machine-aided human translation (e.g. postediting, translation memory, multilingual text processing, etc.). We plan to add material on these topics in the future. In general, we find it necessary to adjust the materials and topics in the course as the MT field evolves, language technologies improve, and new technologies emerge. Although our current MT course represents 7 years of development and refinement, we assume that the course will continue to evolve due to the changing nature of the field.

Bibliography

Arnold, D, L. Balkan, R.L. Humphreys, S. Meijer, L.Sadler (1994). "Machine Translation - An Introductory Guide", NCC Backwell Ltd., Oxford, UK. http://www.essex.ac.uk/linguistics/clmt/MTbook/HTML/book.html

Baker, K., A. Franz, P. Jordan, T. Mitamura and E. Nyberg (1994). "Coping With Ambiguity in a Large-Scale Machine Translation System" *Proceedings of COLING-94*, Kyoto, Japan.

Berger, A., S.A. Della Pietra, V.J. Della Pietra (1996). "A Maximum Entropy Approach to Natural Language Processing" *Computational Linguistics*, Vol. 22. Number 1.

Brown, R. (1996). "Example-Based Machine Translation in the Pangloss System". *Proceedings of COLING-96*, pp. 169-174, Copenhagen, Denmark.

Brown, R. and R. Frederking (1995). "Applying Statistical English Language Modeling to Symbolic Machine Translation". *Proceedings of TMI-95*, pp. 221-239, Leuven, Belgium.

Brown, P., J. Cocke, S.A.Della Pietra, V.J. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roossin (1990). "A Statistical Approach to Machine Translation" *Computational Linguistics*, Vol. 16. Number 2.

Carbonell, J., T. Mitamura and E. Nyberg (1992). "The KANT Perspective: A Critique of Pure Transfer (and Pure Interlingua, Pure Statistics,...)", *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, TMI-92, 225-235, Montreal, Canada.

Collins, B, P. Cunningham, T. Veale (1996). "An Example-Based Approach to Machine Translation." In Proceedings of AMTA-96, pp. 1-13, Montreal, Canada.

Dorr, B. J. (1993). "Machine Translation: A View from the Lexicon", Cambridge, MA: MIT.

Frederking, R., A. Rudnicky, C. Hogan, K. Lenzo (2000). "Interactive Speech Translation in the DIPLOMAT Project". *Machine Translation*, Vol. 15. Number 1-2. Special Issue on Spoken Language Translation.

Frederking, R., A. Black, R. Brown, J. Moody, E. Steinbrecher (2002). "Field Testing the Tongues Speech-to-Speech Machine Translation System" *Proceedings of LREC 2002*.

Goodman, K. and S. Nirenburg (eds.) (1991) "The KBMT Project: A Case Study in Knowledge-Based Machine Translation", Morgan Kaufmann Publishers, San Mateo.

Hutchins, John W. and Harold L. Somers, (1992)

"An Introduction to Machine Translation", Academic Press, San Diego.

Hutchins, John W. (ed.) (2000), "Early Years in Machine Translation", John Benjamins.

Hutchins, John W. (1986) "Machine Translation: Past, Present, Future", Ellis Horwood Limited, England.

Lavie, A. (1996) "GLR*: A Robust Grammar-Focused Parser for Spontaneously Spoken Language." PhD Dissertation. Technical Report CMU-CS-96-126, Carnegie Mellon University, Pittsburgh, PA.

Lavie, A., F. Metze, R. Cattoni, E. Costantini (2002). "A Multi-Perspective Evaluation of the NESPOLE! Speech-to-Speech Translation System." In Proceedings of Speech-to-Speech Translation: Algorithms and Systems Workshop at ACL-02, Philadelphia, PA.

Levin, L., D. Gates, A. Lavie, and A. Waibel (1998). "An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues." In Proceedings of ICSLP-98, pp. 1155-1158, Sydney, Australia.

Levin, L., D. Gates, A. Lavie, F. Pianesi, D. Wallace, T. Watanabe, and M. Woszczyna (2000). "Evaluation of a Practical Interlingua for Task-Oriented Dialogue." In *Proceedings of Workshop on Applied Interlinguas: Practical Applications of Interlingual Approaches to NLP*, Seattle.

Manning, C. and H. Schütze (1999). "Foundations of Statistical NLP", MIT Press. Cambridge, Massachusetts.

Mitamura, T., E. Nyberg, E. Torrejón and B. Igo (1999). "Multiple Strategies for Automatic Disambiguation in Technical Translation" it Proceedings of TMI-99, Chester, UK.

Mitamura, T., E. Nyberg and J. Carbonell (1993). "An Efficient Interlingua Translation System for Multilingual Document Production", in Nirenburg ed. *Progress in Machine Translation*, Amsterdam, IOS Press. Originally published in *Proceedings of Machine Translation Summit III*, 1991, Washington, DC.

Nielsen, Jakob (1993). "Usability Engineering," AP Professional, Boston, MA.

Nyberg, E., T. Mitamura and Huijsen (2003). "Controlled Language for Authoring and Translation" in *Computers and Translation*. Harold Somers (ed.) Benjamins Translation Library.

Papineni, K., S. Roukos, T. Word and W.J. Zhu (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation", In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia.

Reiter, Ehud and Robert Dale (1995). "Building Applied Natural Language Generation Systems," *Natural Language Engineering*, 1(1).

Somers, H. (2001). "Three perspectives on MT in the classroom", MT Summit VIII Workshop on Teaching Machine Translation, Santiago de Compostela.

Tomita, M.(ed.), T. Mitamura, H. Musha and M. Kee (1988). "The Generalized LR Parser/Compiler: Version 8.1 User's Guide", CMU-CMT-88-MEMO. Technical Memo. CMU, Pittsburgh.

Tomita, M. and E. Nyberg (1988). "Generation Kit and Transformation Kit: Version 3.2, User's Manual", CMU-CMT-88-MEMO. Technical Memo. CMU, Pittsburgh.

Trujillo, Arturo (1999) "Translation Engines: Techniques for Machine Translation", Springer-Verlag Series on Applied Computing.

Teaching the automation of the translation process to future translators

Benoît Robichaud and Marie-Claude L'Homme

Département de linguistique et de traduction
Université de Montréal
C.P. 6128, Succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7

benoit.robichaud@umontreal.ca
lhommem@ling.umontreal.ca

Abstract

This paper describes the approach used for introducing CAT tools and MT systems into a course offered in translation curricula at the Université de Montréal (Canada). It focuses on the automation of the translation process and presents various strategies that have been developed to help students progressively acquire the knowledge necessary to understand and undertake the tasks involved in the automation of translation. We begin with very basic principles and techniques, and move towards complex processes of advanced CAT and revision tools, including ultimately MT systems. As we will see, teaching concepts related to MT serves both as a wrap-up for the subjects dealt with during the semester and a way to highlight the tasks involved in the transfer phase of translation.

1. Introduction*

This paper describes several aspects of a course entitled "*Outils informatiques des langagiers*" (Engl. "Computer tools for language professionals") in which a wide range of computer-assisted translation (CAT) tools and other applications – including machine translation (MT) systems – must be covered. The course is offered in all undergraduate translation curricula of the Département de linguistique et de traduction at the Université de Montréal (Canada). The focus here will be on the automation of the translation process and the strategies developed to help students progressively acquire knowledge on translation tools. We begin with fairly simple concepts, principles and techniques of basic tools and move on to the understanding of more complex processes found in advanced CAT and revision tools, and ultimately in MT systems.

As we know, concepts related to MT are not readily accessible, especially for students who do not have a background in linguistics or computing. Nonetheless teaching those concepts may serve both as a wrap-up for the various subjects covered during the semester and as an excellent way to draw attention to the tasks involved in the transfer phase of translation, which is often accomplished intuitively.

We first present the general objectives of the course and discuss some relevant technical details, as well as the academic background of the students. In the remaining sections, we set out the course contents, including the different themes, techniques and tools that are reviewed and taught throughout the semester, giving short examples of exercises and assignments. We close these sections by focusing on the benefits of ending the semester with the presentation of the concept of "fully automatic MT." We conclude the article with a few remarks on the reactions of the students to the course and suggestions for improvements in the future.

2. Course objectives and settings

The main goal of the course is to present a broad range of computer applications that translators will be asked to work with on modern workstations. It focuses on giving trainee translators the means to master the computerized tools in a professional setting, either in translation firms or as freelance workers. It draws heavily on the automation continuum of the translation process from the human's

* We would like to thank Elizabeth Marshman for very helpful suggestions. Thanks also to three anonymous reviewers for useful comments on an earlier version of this paper.

point of view, as pioneered in Kay (1980) and others.[1] Ultimately the course should enable the learners to evaluate the possibilities, limitations and impacts of each tool against the gain it represents for various translation needs and in different settings.

Apart from Master's and Ph.D. levels,[2] the course is compulsory in all translation curricula leading to translation markets. Groups are composed of undergraduate students (certificates and Bachelor of Arts degrees) and graduate students (in a one-year specialization program). Most of the students work in English-French translation, but a growing number of alumni have other working languages (Spanish, for instance).

The course is given in 45 hours over fifteen weeks. Except for two classes devoted to exams, each class is divided equally between a lecture and a hands-on session in a computer laboratory. The laboratory is equipped with workstations on which the various applications are installed. The students are instructed to read some chapters of a reference book (L'Homme, 2000) to prepare for the lectures,[3] and to refer to a dedicated university website for supplementary notes and links (L'Homme and Robichaud, 1999). It is noteworthy that the website also contains practical instructions for the exercises students must complete at the laboratory sessions. This medium has proved to be a very valuable tool for both students and instructors. Students can refer to the website whenever they want and can do the exercises at their own pace. Finally, the evaluation is based on two exams on the material covered during the lectures in addition to five assignments consisting of selected exercises involving the use of specific computer applications in concrete translation situations.

It must be emphasized that "*Outils informatiques des langagiers*" is the only course entirely devoted to computer applications that is offered as part of the translation programs. Hence, it must cover several different topics in a relatively short period of time (see Figure 1). However, an increasing number of professors are becoming aware of the importance of including some applications, and ask students to use them.

Although all of the students are registered in a translation program, their levels of competence and knowledge of the professional world and of the different computer applications are somewhat heterogeneous. Therefore, special attention is paid to developing course content that suits the needs of all the students, whether they are well accustomed to working with computers but are less acquainted with translation practices as such, or have practical experience in translation and are taking the course to remain up-to-date. In all cases, students are warned at the very beginning that the course is neither an introduction to computer use nor to word processing, and they are advised to do extra work in order to catch up if necessary.

3. Contents of the course

Figure 1 gives a general overview of the material covered during the course. The important topics are discussed in detail in the following sections.

- Historical overview of translation and computing with key terminological distinctions such as MT, MAT, MAHT, HAMT, FAHQMT, etc.
- Translation situations, translators' needs and the automation of the translation process
- Computer networking, Internet and telecommunications (basics)
- Word processing ("advanced" functions)
- Translation problems and terminology management (database management systems and terminology management software)
- Text corpora (indexing and searching monolingual concordances)
- Paired text corpora (aligning and searching bilingual concordances)
- Recall tools (translation memories)
- Proofreading tools (spelling and grammar checkers)
- Machine translation

Figure 1: Overview of the contents of *"Outils informatiques des langagiers."*

[1] See discussion in Somers (1999).

[2] Master's and Ph.D. students in translation can take an advanced course entitled "*Traductique*" where the point of view on CAT tools shifts from use to design. Let us mention that advanced computational linguistics courses are offered in graduate linguistics and computer science curricula, where functioning and development aspects of current MT systems are covered in more depth. This division is similar to the one mentioned in Kenny & Way (2002).

[3] Other references of recent introductory textbooks such as Austermühl (2001) and Bowker (2002) are also given.

The first lectures are dedicated to a brief introduction to the historical context of MT and machine-aided translation (MAT) and show how this distinction developed. Several factors that have led to the computerization of the translation profession are then presented and important terminological distinctions are covered to show how computers can be included in the translation process and how they change the way translation is carried out nowadays.

The remainder of the course is devoted to the specific applications with special attention to the order in which they are introduced and practiced. We try to situate the uses of each application in the context of basic but realistic translation tasks (see next sections). The general idea is to begin with relatively simple processing strategies and techniques and move towards more complex ones. The objective is to facilitate the understanding of the overall functioning of a given application and to show how other applications may go further towards fulfilling specific translation needs by automating particular tasks.

For example, during the class on word processing, a few explanations are given on the word count and the search and replace functions. This allows the instructors to introduce basics of character string manipulation and to detail various problems related to word recognition, such as part-of-speech and semantic ambiguities. Once the concept of "ambiguity" is mastered in the context of a search and replace task where the "replace all" function cannot be used satisfactorily, students can easily understand why it becomes a real problem when using a grammar checker or MT software. Another example is the order in which monolingual corpora, aligned corpora and translation memories are introduced. Students learn first how to build text concordances and execute search queries in indexed monolingual corpora. They are taught next how source texts and their translations can be aligned, then also indexed and searched. Last, they discover how sentences can be automatically located and retrieved by translation memory software from its database of aligned texts. Several other examples are given below.

Finally, when teaching the various computer applications, we try to focus on the processing techniques involved and not on specific off-the-shelf software. Students should normally be able to transpose their knowledge to other commercial utilities when they begin working as professional translators.

4. Starting with the translation process

During one of the first classes, we introduce a simplified, seven-stage translation process (see Figure 2). We illustrate how each phase may be automated in some way, along with the diverse tasks that the translator will have to accomplish individually or with coworkers. Throughout the course, this model is augmented and adapted to cover different translation situations.

- Reception of the source language text
- First reading
- Documentary and terminological searches
- Translation *per se*
- Proofreading and correction
- Revision
- Delivery of the target language text

Figure 2: A simplified translation process.

The presentation of this simplified translation process leads us in the following classes to the introduction of the specific software applications and in parallel to the various degrees of processing difficulties that translation automation faces.

5. Basic tools

The first two applications that are dealt with are Internet browsers and e-mail tools. We do not spend much time on these topics in lectures, but introducing them during the first session at the laboratory ensures that all the students will master document downloading and exchanging. This is essential since all assignments are to be handed in as e-mail attachments. This first practical session also covers logins (on computer networks as well as e-mail and course website accounts) and a brief review of the OS environment, file system storage and software accessories such as archiving tools, PDF readers, etc.

The major and most important application that all translators must master at various stages of the translation process presented above is evidently the word processor. It might seem surprising that word

processing is still taught in a course at the university level, but experience has shown us that many of the functions we review are unknown to a majority of the students.

We quickly review basic functions such as word count, search and replace, tables and sorting, etc. These functions are excellent starting points for introducing linguistic and computational concepts such as the internal representations of characters, words and sentences for computer software, related word and sentence segmentation problems, string comparison, and the like. All of these notions will be crucial when we bring in advanced CAT and revision tools. We then move toward functions we view as "advanced" and classify them according to specific tasks. These include format conversions, document management and comparison, hyperlink insertion and management, annotation and tracking changes, macros, autocorrection lists, spell checking and personal dictionary management.[4] Exercises with the word processors are developed and related to the different parts of the translating process involving word processing. A few examples of exercises are given in Figure 3.

- Display and correct spelling errors
- Carry out a word count on a text and find out if the count is similar to that carried out by a translator
- Build a simple macro to manage word count adequately
- Create and sort tables to organize terminological data
- Standardize terminological choices both in texts and terminological tables with the search and replace function
- Automate the creation of a terminological table with a macro
- Compare two versions of a document
- Save a file in different exchange formats
- Organize the desktop to work efficiently with many open files and/or applications

Figure 3: Exercises with word processors.

The next important task of the translation process that we cover is terminology management, for which we use database management systems.[5] Of course, many translators may be able to afford and/or will be required to work with dedicated terminology-management systems (TMS). Nevertheless, we feel that it is important to renew with basic concepts such as records, fields and data types, relating them to a central concept, namely the "query." We use a generic database management system for this purpose and show students how to build terminology databases from scratch. The links established with the query permit us to introduce many new concepts.

First, we cover basic elements of set theory, such as union, intersection and complement. Students are then expected to be able to handle Boolean operators in various types of queries. Secondly, we present the notion of character masks such as those used in regular expressions and practice the use of wildcards to perform stemming and other kinds of searches. Third, based on the previous elements, we introduce the concept of adjacency (relative proximity and order) of words and practice the extraction of collocations from certain text fields. The mastering of those concepts will be very important for later classes devoted to text concordances.

Lastly, we briefly review record structures and basic retrieval functionalities[6] that are commonly offered by commercial TMS and compare them (price, learning curve, flexibility of importing and merging, etc.) with generic database management system that are now universally distributed with office suites. Different exchange formats that are presently available or emerging are also discussed at this point.

Exercises are related to the design, construction and querying of small terminological databases. Capturing, importing and exchanging terminological data are also thoroughly practiced. Examples of exercises students are asked to do at this time are given in Figure 4.

[4] Grammar checking is postponed until later on as we use stand-alone software and present the functionality in the context of proofreading and revision. During the class on word processing, we show students how spelling errors can be displayed automatically on screen or browsed one by one after launching the "Spelling and Grammar" function.

[5] Public term banks such as *TERMIUM®*, *EURODICAUTOM* and the *Grand dictionnaire terminologique* are covered in an earlier course exclusively devoted to documentation and terminology searches.

[6] Terminology extraction and pre-translation functionalities are also discussed later with advanced CAT tools.

- Design and build databases of varying complexity from scratch
- Build appropriate front-ends to display, capture and manage terminological data
- Import lists (created with the word processor or found on the Internet) automatically
- Perform different types of queries in databases
- Link different databases
- Organize and export selected contents of databases

Figure 4: Exercises with database management systems.

One of the important abilities for translators these days is undoubtedly to know how to build and query text concordances locally for terminology or documentation purposes. We introduce the distinctions between text collections and corpora (Bowker, 2000) along with the functioning and use of different types of monolingual concordance software, and we again address the concepts of indexation and searching. During the laboratory session, students learn how to search for strings, words and combinations of words in technical text corpora. The set notions, as well as the wildcard and word adjacency concepts discussed above are of primary importance here since the results of simple queries may be awfully noisy or on the contrary extremely slim.

- Build concordances of monolingual text corpora
- Perform various types of queries to find words, parts of words, word combinations and collocations
- Find ways to locate information that may help solve specific translation problems
- See and understand how textual data provides information that is not normally compiled in standard reference works such as dictionaries or term banks
- Place appropriate information retrieved in terminological databases

Figure 5: Exercises with concordancers.

The results of the various searches carried out as exercises are placed in new, manually created databases specifically designed for the different kinds of terminological problems that are being addressed. A few examples of tasks students must accomplish are given in Figure 5.

We next turn to bilingual concordances and explain techniques used to build bilingual corpora from source texts and their translations stored on a local workstation. Students must first create very short aligned corpora with manual and semi-automatic alignment methods using hyperlinks and a word processor macro. Then they learn to import and exploit the resulting corpora within the generic database management system. They are next taught to use and evaluate automatic alignment methods such as the ones found in Trados' *WinAlign* and Terminotix' *Logiterm* commercial applications.[7] We finish by introducing the RALI large bilingual databases and review the functionalities implemented therein to mask inflection and search for ellipses.[8] Examples of exercises are given bellow.

- Build a bilingual corpus using different methods (first manually with hyperlinks, second with a word processor macro, and third automatically with alignment tools)
- Import a bilingual aligned corpus into a generic database management system and perform queries
- Perform queries within the interface of alignment tools that support this
- Find texts on the Internet that could lend themselves to automatic alignment
- Perform queries in large aligned corpora

Figure 6: Exercises with aligned corpora.

6. Advanced tools

Once all sorts of concordance application software along with relevant functionalities have been thoroughly studied, we introduce recall tools as the first type of advanced CAT tools. Those involve higher-level automation of the translation process in some well defined settings such as updated versions of previously translated texts, highly repetitive technical texts, etc. We show students how these new tools are designed to automatically reuse (or recycle) segments of previously translated text stored in their databases.

First, the translation memory (TM) is introduced as a set of textual databases (viewed as a sort of aligned bilingual corpus) supplemented by a pilot (control) program. Detailed explanations are given on the functioning of the pilot that automatically

[7] See www.trados.com and www.terminotix.com.
[8] See www.tsrali.com.

divides the source text into segments (usually sentences), then retrieves and displays the "best translation solutions" from its databases. A relation is established with aligned corpora that have been broadly dealt with in previous classes. Further explanations are given on the specific contexts of use and functionalities that are commonly offered (pre-analysis of the source texts to evaluate their level of repetition and the database's applicability, integration into word processors, manual look-ups and updating of the databases, etc). More importantly, we review the different strategies the software uses for establishing exact and fuzzy matches along with methods for fine-tuning the best type of match according to translation situations.

Secondly, we present a functionality usually found in TM or TMS, which we refer to as a "vocabulary translator." The tool is described as a computerized bilingual dictionary that is also enhanced by a pilot control program. This time, the pilot is capable of searching for and replacing words, terms or idioms automatically with their lexical or terminological equivalent in the text to be translated.[9] The students are warned that the result is a partially translated text (as opposed to aligned ones seen previously), which has to be lexically verified and rephrased at the syntactic level in a post-editing phase. A links are established with terminological databases and dictionary look-ups that have also been dealt with in previous classes. Explanations are again given on the specific contexts of use as well as on the regular functionalities that are expected from this second kind of recall tool. Examples of exercises given to the students are shown in Figure 7.

- Translate an updated version of a text using a (demo version) translation memory interfaced with the word processor
- Search and modify the content of the translation memory database manually
- Run a (demo version) "vocabulary translator" on a source text and rewrite the resulting text into an acceptable translation

Figure 7: Exercises with translation memory and vocabulary translator applications.

[9] This processing is sometimes referred as "active terminology recognition" or "pre-translation."

If time allows, we turn next to terminology extraction.[10] We present software applications that extract and produce lists of lexical or terminological items found in source texts. Context of use and expected functionalities (sort and synthesis options) are presented along with a discussion on the combination of various strategies implemented in TMS that perform this task automatically (segment repetition, syntactic patterns, statistical collocations, etc.). Links are made with word segmentation and recognition, dictionary look-ups and corpus indexing techniques that were dealt with in previous classes. Examples of exercises are shown in Figure 8.

- Extract terminological content from a technical text and present different lists using the sort and synthesis options
- From the same source text, evaluate the appropriateness and rank of the various items found in the lists
- Place genuine terminological items in previously created terminological databases
- Explain why some terminological items of the source text have not been placed or have been misplaced in the lists

Figure 8: Exercises with terminology extraction software applications.

As seen in the translation process presented above, proofreading is still a very important step in producing a high-quality translation. While translators are skilled writers, for reasons that we will not review here, they may sometimes miss errors and typos made during the translation *per se*. Moreover, other applications involved at previous stages of the translation process may also have introduced errors.

Next to be introduced are tools such as spelling and grammar checkers. We briefly review the historical context and the technical changes that have been made from spelling correction to full grammatical checking. We next present how such tools can carry out "parsing," described as a sort of symbolic calculus on syntactic structures and

[10] As mentioned above, many concepts, techniques and tools have to be covered during the course. From time to time, choices have been made by instructors to summarize this part in the context of terminology management in the interests of developing a deeper understanding of the other topics.

relying on grammatical and lexical knowledge. At this time we bring in structural ambiguities (related to the part-of-speech and semantic ambiguities discussed earlier) and how they can also lead to erroneous results. We then introduce notions such as "noise" and "silence" (in this context, overcorrection and undercorrection) to cover the different types of difficulties and limitations involved in using the tools. Finally, other technical topics such as integration into word processors and generally expected functionalities are covered. Examples of exercises students must complete for this class are shown in Figure 9.

- Correct a list of incorrect sentences with the grammar checking functionality integrated in a word processor
- Correct the same list of sentences with a stand-alone grammar checker
- Compare corrections of both grammar checkers by highlighting "noise" and "silences" for each of them.
- Identify the problems involved when the corrections are erroneous (part-of-speech and structural ambiguities, etc.)

Figure 9: Exercises with spelling and grammar checkers.

The semester normally ends with a class on MT systems, which enables us to come back to important distinctions between MAT and MT. The ideal contexts in which MT systems may be used and the impact of these uses on the translation profession are also briefly discussed.

We next describe basic concepts of well known analysis-transfer-generation approaches of the kind introduced by Vauquois (1968) as well as related principles of new approaches such as example-based MT. Links are established with several of the concepts and much of the information introduced in connection with previous processes, and the focus is then narrowed to the difficulties associated with the transfer phase in particular. At this stage of the course, students have acquired the necessary background to understand the most important steps of the automation of the translation process and the difficulties and limitations that fully automatic MT systems may face. They also learn why those errors and limitations may compound each other, leading in a domino effect to gross translation mistakes. Examples of exercises are shown in Figure 10.

- Translate a short text with some MT systems (one on the workstations, the others available on the Internet)
- Compare machine outputs and for each classify translating errors into known categories (part-of-speech and structural parsing ambiguities, semantic and structural transfer ambiguities, word order and agreement errors, etc.)
- Choose the best output and rewrite it into an acceptable translation

Figure 10: Exercises with MT systems.

7. Concluding remarks

This course has been given for nearly a decade and has evolved considerably since the first time it was given. Certainly it is important to take into account advances in the field of translation computerization and remain up to date. However, the major motivation for the changes the course has undergone was to build a better understanding of the automation process itself. Before redesigning the course we used to teach the various applications in a modular fashion and failed to establish links between them. Students were simply learning to manipulate separate functions and did not have a general perspective on the field.

Using the translation process and basic concepts related to natural language processing as the backbone of the course appears to be the best solution to organize the material and to help students acquire the fundamentals in the field. A simple example will serve as an illustration of this last observation. At the end of the semester, during the sessions on proofreading and MT, most students are impressed by everything that is involved in producing the output and stop focusing exclusively on errors. They can also develop a critical evaluation and pinpoint specific problems instead of simply stating that the output results are "bad."

We think it is worthwhile to keep moving in this direction and it seems possible to continue incorporating easily new CAT applications as well as MT systems in this overall organization.

8. References

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

Bowker, L. (2000) 'Towards a Methodology for Exploiting Specialized Target Language Corpora as Translation Resources.' *International Journal of Corpus Linguistics*, 5, pp. 17-52.

Bowker, L. (2002) *Computer-aided Translation Technology. A Practical Introduction*, Ottawa (Canada): University of Ottawa Press.

EURODICAUTOM (2003) *The European Commission's multilingual term bank.* European Commissions. (http://www.europa.eu.int/eurodicautom)

Le Grand dictionnaire terminologique (2003) *La base de données terminologiques de l'Office québécois de la langue française.* Gouvernement du Québec (http://www.granddictionnaire.com)

Kay, M. (1980) 'The Proper Place of Men and Machines in Language Translation.' Report CSL-80-11 Xerox Palo Alto Research Center, Palo Alto, CA. Reprinted in [1997] *Machine Translation*, 12, pp. 3-23.

Kenny, D. and A. Way (2001) 'Teaching Machine Translation & Translation Technology: A Contrastive Study.' *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, pp. 13–17.

L'Homme, M.-C. (2000) *Initiation à la traductique*, Brossard (Canada): Linguatech.

L'Homme, M.-C. and B. Robichaud (1999) "*Outils informatiques des langagiers*," Course website, Université de Montréal.

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

Somers, H.L. (2001) 'Three Perspectives on MT in the Classroom', *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, pp. 25–29.

TERMIUM® (2003) *La base de données terminologiques et linguistiques du gouvernement du Canada*. Gouvernement du Canada. (http://www.termiumplus.com)

TS-RALI (2003) *TransSearch search tool on the Canadian Hansard bilingual corpus.* (http://www.tsrali.com)

Vauquois, B. (1968) 'A Survey of Formal Grammars and Algorithms for Recognition and Transformation Machine Translation.' *IFIP Congress-68*, Edinburgh, pp. 254-260.

Prolog models of classical approaches to MT

Harold SOMERS

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

Abstract

This paper describes a number of "toy" MT systems written in Prolog, designed as programming exercises and illustrations of various approaches to MT. The systems include a dumb word-for-word system, DCG-based "transfer" system, an interlingua-based system with an LFG-like interface structure, a first-generation-like Russian-English system, an interactive system, and an implementation based on early example-based MT.

1. Introduction

There is something of a consensus in the recent literature on "MT in the Classroom" that distinguishes the teaching of MT to different target students, notably Computer Science (CS) and Computational Linguistics (CL) students vs. trainee translators (cf. Kenny & Way, 2001; Somers, 2001) among others. The present paper discusses the development of "toy" systems (cf. v. Hahn & Vertan, 2002), specifically written in Prolog, developed with the aim of providing relatively complex natural-language programs for students of Prolog and, secondarily, to provide demos of various approaches to MT. A number of different of languages are illustrated, usually for no other reason than variety. Some of the systems described here were developed with the help of students at UMIST.

An obvious and reasonable initial question might be why working on model implementations in Prolog might be useful, and to what sort of student? Way (2002) describes how his students undergo three years of training in CL including programming in various languages. he states that these students "may find themselves in the position of implementing changes to current systems, or indeed developing new ones" (p. 54). He describes exercises which involve Prolog programming within a framework based on Eurotra's E-framework (Bech & Nygaard, 1988), and it might be an interesting point of discussion for both Way and the current author whether Prolog is the most appropriate vehicle for this activity, given its lack of status as a programming language of choice for a real industrial application. We will leave this discussion for another time!

2. Similar approaches

The (relatively restricted) literature describing teaching of MT aimed at CL and CS students can be divided into those that use standard programming languages, and those that have developed more task-specific tools.

We have described Way's (2002) approach above. The students of v. Hahn & Vertan have arguably an even more difficult task, as they are left to devise on their own the basic architecture. From their discussion it appears that the only tools provided beforehand for the students is a corpus of test sentences, and a full-form lexicon in an XML-like formalism. No mention is made of the choice of programming language.

Amores (2002) describes Xepisteme, a tool for developing LFG-based MT systems. Amores stresses that the transparency of the system means that the user requires no programming skills theough he admits that an appropriate specification language must be learned, entailing the "usual difficulty" (p. 65), and indeed the examples of rules and so on that are shown suggest that this specification language is effectively a high-level task-specific programming language. Nevertheless, the tool also has a graphical interface that displays c- and f-structures, and the illustration is reminiscent to this author of the Metalshop tool that developers of the Metal system had available in the 1980s, as illustrated in Hutchins & Somers (1992: 264ff),and still available to users of the commercial *T1* system.

Sheremetyeva (2002) illustrates the use of the "developer tools" part of the APTrans system. It si not always clear from the paper how all the tools are used, the impression is that the ideal user will be a highly trained computational linguist. But no programming skills as such seem to be implied.

3. Systems that students can develop

3.1 A really dumb system

The simplest of MT models, if it can be called that, is one which translates word-for-word. This is very easy to implement in Prolog, which has good list-handling facilities. In fact, it is basically a one-predicate program (1) accompanied by a lexicon.

```
translate([],[]).
translate([S1|SRest],[T1|Trest]) :-
        tx(S1,T1),
translate(Srest,Trest).
tx(cat,chat).
tx(dog,chien).
tx(the,le).
tx(chased,chassa). % etc.
tx(X,X).
```

(1) *Word-for-word replacement model*

Obviously, this program will result in crude translations, but can serve as a starting point to bring home to students how inadequate this naïve approach is, in a way similar to the method described by Pérez-Ortiz & Forcada (2001), and can be used to invite students to identify what needs to be added. Even with a program of this simplicity, students can surprise you. One student who was brave enough to undertake translation between two foreign languages (French and German) needed much convincing that his rules mapping *le* onto *der* and *la* onto *die* left something to be desired.

3.2 A Prolog-like model: DCG-based transfer

Since we are programming in Prolog, an obvious model is one which takes best advantage of the most appropriate features of Prolog. That an MT system implemented in Prolog does not necessarily do this is well illustrated by McCord's (1989) LMT which includes code like that shown in (2), which is evidently a sequence of actions that could just as easily be programmed in any procedural language.

Prolog provides the DCG formalism (Pereira & Warren, 1980) which of course originally developed out of work on MT (cf. Colmerauer & Roussel, 1993). Although a very flexible formalism, it is usually associated with phrase-structure grammars, and this is the approach for our next model, also found in Gal et al. (1991:181ff). All other things being equal, DCGs can be run as parsers or generators, and so a system which links two DCGs will be reversible. This model consists of linking the source- (SL) and target-language (TL) DCGs with a transfer component which maps the SL phrase-structure trees onto those appropriate to be input to the TL grammar. The system is reversible if "Prolog escapes" in the DCG are used judiciously, for example for agreement and subcategorization. (3a) shows the top-level predicate, (3b) some example rules. (4) shows the transfer "formalism". Note that the DCG rules must be distinguished (e.g. `snp` vs `tnp`), but the structure is shared. The `tnp` rule shown here illustrates one way to do agreement.

```
nt(F,G,H,I,J,syn(Lab,B1,Mods),U,Z) :-
  copylabel(nt(F,G),nt(F1,G1)),
  bbrackets(B,U,V),
  preconj(PC,Mods,Mods1,V,W),
  ntbase(F1,G1,H,I,J,B,SynO,W,X),
  ntconj(F1,G1,F,G,H,I,J,PC,SynO,
            syn(Lab,B1,Mods1),X,Y),
  ebrackets(B1,Y,Z).
```

(2) *Example from LMT (McCord, 1989:37)*[1]

```
%(a)
translate(SText,TText) :-
                ss(Sstruc,Stext,[]),
                transfer(SStruc,TStruc),
                ts(TStruc,Ttext,[]).
%(b)
ss(s(NP,VP)) --> snp(NP), svp(VP).
tnp(np(Det,N,Adj)) -->
            tdet(Det), tn(N),tadj(Adj),
            {agree(Det,N,Adj)}.
```

(3) *DCG-based transfer system.*

The transfer component works top-down on the tree-structure, applying the most specific rules first, with a default of structure preservation. The examples show complex structural transfer for head-switching or "idioms" , and TL word selection based on category.

3.3 An LFG-like interlingua model

Another model that we present, again based on DCGs but this time without the explicit transfer phase, builds predicate–argument structures not

[1] The Prolog syntax has been harmonized with the syntax used elsewhere in this paper.

```
%(a)
transfer(vp(v(swim,SVF),
            pp(p(across),NPS)),
         vp(v(cruzar,TVF),
            np(NPT),adv(natando))) :-
         transfer(SVF,TVF),
         transfer(NPS,NPT).
%(b)
transfer(vp(v(get_up,SVF),adv(early)),
         vp(v(madrugar,TVF))) :-
         transfer(SVF,TVF).
%(c)
transfer(n(book),n(libro)).
transfer(v(book,SVF),n(reservar,TVF)) :-
                      transfer(SVF,TVF).
```

(4) *Transfer rules*

The transfer rules illustrate (4a) *swim across* → *cruzar natando*,(4b) *get up early* → *madrugar*, and (4c) *book* translated as *libro* (noun) or *reservar* (verb).The structures assume that verb features(`SVF`, `TVF` etc.) have been percolated, and are also subject to transfer.

unlike LFG f-structures, taking advantage of Prolog's straightforward unification (Gal et al., 1991:180). (5a) shows an NP rule that builds a feature structure, and (5b) the kind of structure that can be built. This is then transformed into a TL structure via rules (5c) which map SL attribute–value pairs which may be lexical or otherwise.

4. Return to Georgetown

The remaining systems to be described are more for demo purposes. The first[2] was developed more for fun and illustration than as a model of how to do MT in Prolog. I wanted to replicate the "first generation" design as described in sources such as Hutchins & Somers (1992:72), in which translation consists of morphological analysis, bilingual dictionary look-up and then "local reordering" which is a not particularly systematic attempt at handling structural and other divergences between the two languages. True to the spirit of the first generation, I decided to handle Russian–English, and, not knowing much Russian decided to work through that old stand-by *Teach Yourself Russian* (Fourman, 1943). The system covers the first 15 lessons, and has a vocabulary of 820 words.

The translation process begins with morphological analysis of the Russian input which, like in the original, is in transcription. This in turn introduces problems of ambiguous letter sequences which would evaporate if input

```
%(a)
% NP ->     det     n
%         ↑.num=↓ ↑=↓
%         ↑.det=↓ ↑=↓
np(np(Det,N),Fnp) --> det(Det,Fdet),

{unify([],[num:Num,det:Fdet],F1),
           extract(Fdet,num:Num)},
   n(N,Fn), {unify(F1,Fn,Fnp)}.
%(b)
[pred:eat(subj,opt(obj),
 tense:past,
 num:plur,
 subj:[pred:boy,
       num:sing,
       det:[num:sing]],
 obj: [pred:cake,
       num:plur,
       det:[num:plur]]
]
%(c)
xlex([pred:boy],[pred:ragazzo,gen:masc])
.
xlex(det:[num:X],det:[num:X,gen:Y]).
xlex(eat,mangiare).
```

(5) *LFG-like interlingua system.*

The rule in (5a) assumes a predicate `unify/3` *which succeeds if its third argument is a unification of its first two, and* `extract/2` *which finds the given attribute–value pair from a feature structure. (5b) shows the structure for* The boy ate the cakes*, and (5c) some rules for Italian, in which a feature for gender is introduced. The TL grammar will unify the NP structure and pick up the appropriate gender for the determiner. Notice that* `xlex/2` *rules can specify feature structures, attribute–value pairs or atoms.*

was in Cyrillic.[3] Russian morphology is quite rich, and the analysis creates a "word record" which indicates the position in the sentence, the word and underlying stem, and any grammatical information picked up from either the dictionary or the morphology.

The word is then looked up in the dictionary, which generally pairs the Russian and English words one-to-one. Sometimes however the dictionary lists alternatives, in which case an explicit lexical disambiguation procedure is triggered. Otherwise, ambiguities can have been introduced by the morphological component. This procedure, very much (it is hoped) in the spirit of the first-generation systems, can look at the details associated with any word to the right or left of the word to be disambiguated, or within a specific window. The system does not build any structure as such, so we have to rely on rules like the ones exemplified in (6).

[2] Nicknamed the "Georgetwon system", due to my inability to type that word correctly.

[3] I did develop a version of the system that allowed input in Cyrillic.

(6) a. *говорить* → *speak* if any word (later) in the sentence begins with a capital letter; else *say*
b. *клуб* → *cloud* if there is a preposition up to three words before it, and the sentence contains the verb *fly*; else *club*.
c. *моеи* → genitive of *моя* 'my' if followed by a genitive noun; else imperative of *мыть* 'wash'
d. the adjective endings *-им* and *-ым* can indicate instrumental singular masculine or neuter, or dative plural any gender: look for a preceding preposition which governs one or other case, or look for a following noun of the appropriate gender and case.

The "restructuring" phase is similarly fairly *ad hoc* in nature, and includes rules such as those illustrated in (7). Some are quite general, others specifically mention lexical items.

(7) a. The neuter nominative singular short form of an adjective is an adverb if the verb is not *be*
b. If there is no verb, translate *y* + N1(acc) N2(nom) as N1(nom) + *have* + N2(acc)
c. If there is no verb, insert be after the first pronoun, or before an adverb, or as the 2nd word.
d. Insert indefinite article on singular subject, unless verb is inverted.
e. *искать* + acc → *look for*
f. insert *with* if instrumental is not preceded by preposition

A final pass handles some trivial aspects of English morphology.

The system works well in the sense that it accurately reproduces the kind of translation quality that you might expect with this approach – see (8) for some examples (English output only). Many sentences are translated quite well. Many more are understandable but slightly odd, and there is a pervading Russian-ness about the output (read the examples with a cod Russian accent). And some are just gibberish. I am sorry to say I did not try it with *Душа готова, но плоть слаба*.[4]

a coast is not big.
I don't know to speak Russian.
I love Russian popular songs.
to our street be always many avtomobilej.
I didn't have the letter from my friend.
their conversation lasted whole hour.
we sent own baggage with the fast train.
hand over me, please, newspaper which lies by you.
why you don't lay their to the shelf.
this knigam is here not a place.
I want to the tea.
we saw some picture young artist, about who you read in the newspaper.
chji of the picture you saw yesterday to the exhibition.
my mother ask me to bring her didn't big carpet.
nashi friends came to a harbour to see off nas.
want cigarette?
in the evening after the work I rest.
give to the ill woman of the milk.
he didn't listen to the teacher's explanation.
my son doesn't understand this simple rule because he not listened to the teacher's explanation.
does your sister speak Russian ? no, she doesn't say, but she understands all, what she reads.
a brother's English newspaper is here.
from the coast of the sea to our house is not remote.
at-home is your father ?
do you wish the plate of the meat ?
a left sleeve of the new woollen dress too(much) is long and narrow.

(8) *Example translations from the Georgetwon* [sic] *system.*

5. Interactive system

Another area of interest was interactive MT, which "enables direct involvement on-line during the course of translation" (Hutchins & Somers, 1992:77). Obviously it would be a major undertaking to try to emulate the sophisticated interactive interfaces available with commercial systems, so the aim with this system was more to explore some issues which had become apparent at the time when I was associated with the Ntran project (Whitelock et al., 1986), and in my own even earlier work (Somers & Johnson, 1979). These issues are that the system must know not only when to interact, but also how. If the system interacts whenever it meets a problem, from the user's point of view the interactions may seem disjointed and illogical, since they will be following the system's "flight-path". So for example, it may do lexical disambiguation for various parts of the SL text, then "come back" to do some syntactic disambiguation, and then return for a third time to the "same" problem if there is a question of TL lexical choice. On top of this, there may be a tension between the system's

[4] 'The spirit is willing but the flesh is weak.'

need to translate the current text, and its "learning" function whereby it tries to update its dictionaries (and grammars?): the tension arises because of the potential conflict between the answer to the immediate question, and the answer to a more generic question.

Although the architecture of the system[5] is not the primary interest, some care was taken over it. The system has as its basic data structure a classical chart (Kaplan, 1973) with fairly simple feature bundles on the arcs. The system was developed for French–English translation with, as a reference corpus, a small set of sentences from a corpus of Swiss avalanche warning bulletins.[6] The user is assumed to have some knowledge of both languages though in fact since only interactions during analysis were developed, one could claim it to model a user who knows only the source language.[7] The system interacts with the user at three points: after morphological analysis, during syntactic analysis to resolve category ambiguities, and again later to resolve attachment ambiguities; interaction during lexical transfer was planned but unfortunately never developed.

The morphological analysis looks only at suffixes, and operates in a fairly standard rule-based manner. Since our focus is how an interactive system should work, unknown words are of particular interest: we want our model to make whatever inferences it can from morphological analysis. In the case of an unknown word, all segmentations are tried and then these must be presented to the user. (An alternative would be to keep the alternatives on the chart and wait to see if syntactic analysis ruled out the wrong solutions.) French morphology is quite rich, particularly in adjective and verb paradigms: this richness involves not so much a large range of inflections, but a large variety of interactions between stems and endings. Supposing, for example, that the word *basses* was not in the lexicon. According to our rules, there are 21 different interpretations of this string, assuming it is inflected (9).

stem	cat	gen	nmbr	para
basse	n	masc	pl	
basse	n	fem	pl	
basses	adj	masc	sing	1
basses	adj	masc	sing	2
basses	adj	masc	sing	6
basses	adj	masc	sing	11
basses	adj	masc	sing	15
basses	adj	masc	sing	18
basse	adj	masc	pl	1
basses	adj	masc	pl	2
basse	adj	masc	pl	3
basse	adj	masc	pl	6
basses	adj	masc	pl	18
basses	adj	fem	sing	18
bass	adj	fem	pl	1
bass	adj	fem	pl	2
basse	adj	fem	pl	3
basx	adj	fem	pl	4
bas	adj	fem	pl	6
bas	adj	fem	pl	15
basses	adj	fem	pl	18

(9) *21 possible interpretations of* basses.

From a user-interaction point of view, it would clearly be impractical to present all these alternatives in one menu, so the program works out how best to reduce the list by interacting feature by feature: Is it a noun or adjective? Is it masculine or feminine? Is it singular or plural? The correct answers (adjective, feminine, plural) in this case would reduce the 21 possible solutions to seven, at which point an interaction such as shown in (10) can take place.

```
basses
a. stem: bass, paradigm: 1
b. stem: bass, paradigm: 2
c. stem: basse, paradigm:3
d. stem: basx, paradigm: 4
e. stem: bas, paradigm: 6
f. stem: bas, paradigm: 15
g. stem: basses, paradigm: 18

enter letter corresponding to
choice
enter ? for help
enter * to display context
```

(10) *Interaction to disambiguate* basses.

The three last options are presented in all interactions. The "context" option shows the whole sentence, since the interpretation might depend on this. The "help" option in this case explains the paradigm codes, as in (11).

The first syntactic pass sometimes identifies sequences of words which are ambiguous, for example *sont tombés*, which could be copular + adjective 'are fallen' or perfect tense of *tomber*

[5] The development of the morphological interaction part was carried out by Gillian Chamberlain for her 1994 MSc dissertation.

[6] The corpus was obtained from colleagues at ISSCO, Geneva – cf. Bouillon & Boesefeldt (1991).

[7] The fact that the source language is French but the interactions are presented in English is a superficial anomaly.

```
************** HELP MENU ***************
KEY TO OPTIONS: INFLECTION OF ADJECTIVES

1. inflects like "GRAND":      -s -e -es
2. inflects like "GRIS":       -  -e -es
3. inflects like "ROUGE":      -s -  -s
4. inflects like "COURAGEU|X": -x -se -ses
6. inflects like "BO|N":       -s -ne -nes
15. inflects like "GROS":      -  -se -ses
18. invariable, e.g. compass points
****************************************
```

(11) *Help text for adjectives.*
This is canned text but note that only the paradigms mentioned in (10) are explained.

'have fallen'. It is very difficult to know how to ask a user to disambiguate this kind of thing, and we have taken the somewhat unsatisfactory step of presenting the two grammatical analyses more or less "raw" (12).

```
sont tombés
a.
vg([v(tomber,pl,perf(pres),masc,pos)])
b. vg([v(être(tombé),pl,pres,masc,pos)])

enter letter corresponding to choice
enter ? for help
enter * to display context
```

(12) *Interaction to disambiguate* sont tombés.

Of more interest is the second interaction after syntactic analysis, which deals with PP-attachment and coordination. The difficulty here, as we discovered with Ntran (see above) is to frame a question that makes sense even if the answer is "No". Whitelock et al. (1986) illustrate the difficulties in using "natural metalanguage" or "disambiguating paraphrases" to frame interactions (13).

Our solution is to take the lexical heads of the phrases to which the PP might attach and simply present them in a list. When the context is shown close by, this may be an effective method, at least for a user with some "feel" for linguistics. Note that apart from the word *attach*, no linguistic jargon is used (14).

The user can always elect to pass over any interaction. Because the system does not allow criss-crossing attachments, some attachments can be resolved automatically, as a result of other resolutions. For example, in the case in (14), if *des Alpes* is attached to *sont tombés*('fallen from the Alps'), then only options *a* and *f* would be valid choices.

Of all the toy systems, the interactive one is perhaps the least satisfactory as a model, though it does illustrate some difficult questions.

```
"I saw a man in the park"
  1. The action [saw] takes place [in
       the park]
  2. [a man] is [in the park]
"This module provides the interface to
the system"
  1. The action [provides] takes place
     [to the system]
  2. [the interface] is [to the system]

  1. The interface to the system is
     provided by this module.
  2. This module provides the system
     with an interface.
```

(13) *Examples of interaction*
From Whitelock et al. (1986:333). While the first (from Tomita, 1985) works well, with another text the result can be simply absurd. The problem with the third case is finding a set of rules that can correctly generate the paraphrases.

```
"60 à 80 cm de neige sont tombés de
samedi à mardi matin sur le versant
nord et la crête des Alpes au dessus de
2000 m."
Does the PP "au dessus de 2000 m"
attach to:
        a. tomber
        b. samedi
        c. matin
        d. versant
        e. crête
        f. Alpes
```

(14) *Example of PP attachment disambiguation*. [8]

6. Example-based MT

Our final model is a re-implementation of one of the first EBMT systems, the original ATR system (Sumita et al., 1990) handling Japanese adnominal noun phrases of the form A *no* B.[9] These structures are notoriously difficult to translate, the default rendering B *of* A is appropriate less than 40% of the time. Some examples are shown in (15) (from Sumita et al., 1990:207).

Like the original, our model handles translations in the domain of conference registration. It has a database of previously translated examples of A *no* B structures, in fact a subset of ATR's own corpus. The system has a

[8] The text reads '60 to 80 cm of snow fell from Saturday to Tuesday morning on the north face and the crest of the Alps above 2000m.'
[9] This system was programmed by Rachel Patterson for her 1994 MSc dissertation.

(15) a. *yooka no gogo* the afternoon of the 8th
b. *kaigi no sankairyoo* the application fee for the conference
c. *kyooto deno kaigi* the conference in Kyoto
d. *isshukan no yasumi* a week's holiday
e. *hoteru no goyoyaku* the hotel reservation
f. *mitsu no hoteru* three hotels

vocabulary of just over 300 words. The database contains 250 examples, stored as triples consisting of the Japanese text (romanized), the translation, and a coded indication of the target structure, e.g. `BofA`, `AB`, `BtoA`, and so on.

As in the ATR original, the input is matched against the database of examples, and the best match is used as a model for the translation. The matching procedure involves a distance measure based on proximity in a thesaurus. The thesaurus is a shallow (four-level) hierarchy of about 120 domain-specific primitives. A small part of it is shown in (16). All vocabulary items are identified with one thesaurus term, as illustrated in (17) (the first argument is the semantic marker).

```
root
..actions
   ....travelling
   ......travel,sightseeing,sport,activities
   ....booking
......application,registration,reservation,ca
       ncellation,reception,arrangement
   ......attendance
   ....study
   ......research
   ..objects
%etc.
```

(16) *Excerpt from semantic hierarchy*

```
x(sport, tenisu,  tennis).
x(reception, uerukamu, welcome).
x(arrangement, enjo, support).
x(arrangement, 'un-ei', steering).
x(study, kenkyuu, research).
x(research, kagaku, science).
```

(17) *Examples of dictionary entries*

The distance measure is defined as in (18),

$$(18)\quad d(I,E) = \sum \left(sd(I_i, E_i) \times w(I_i, E_i) \right)$$

where *sd* is the semantic distance between I_i and E_i, the corresponding words in the input and example, and *w* is a weighting which reflects the frequency with which the same pattern is used when I_i is translated as E_i. *sd* is given as the level of the most specific common abstraction of the two terms, divided by the depth of the thesaurus, always 4 in our case. *w* is defined as in (19).

$$(19)\quad w = \sum \left(\frac{|patt(I,E,j)|}{N(I,E)} \right)^2$$

where *patt*(*I*,*E*,*j*) is the number of examples where *I* is translated as *E* with pattern *j*, and *N*(*I*,*E*) the total number of examples where *I* is translated as *E*. The idea behind this weight is that if, when *I* is translated as *E*, a variety of patterns is used, then *I* should be less influential in the choice of translation pattern.[10] These weights can be precompiled.

Let us work through an actual example. Suppose we want to translate *rondon no ofisu* (lit. 'London ADN office'). The semantic feature associated with *rondon* is `city`, while *ofisu* is `branch`. Among the examples we look at are (20a–c). The calculations of the distance scores shown in (20).

```
(a) tookyoo no hoteru 'my hotel in Tokyo'      BinA
(c) oosaka no kaigi 'the Osaka conference'     AB
(b) nihon no daigaku 'a Japanese university'   ^AB
   sd(rondon,tookyoo)                  = 0.00
   sd(ofisu,hoteru)                    = 0.25
   w(tookyoo,Tokyo)                    = 1.00
   w(hoteru,hotel)                     = 0.28
   d = (0.00x1.00) + (0.25x0.28)       = 0.07
   sd(rondon,oosaka)                   = 0.00
   sd(ofisu,kaigi)                     = 1.00
   w(oosaka,Osaka)                     = 1.00
   w(kaigi,conference)                 = 0.15
   d = (0.00x1.00) + (1.00x0.15)       = 0.15
   sd(rondon,nihon)                    = 0.25
   sd(ofisu,daigaku)                   = 0.25
   w(nihon,Japanese)                   = 1.00
   w(daigaku,univ.)                    = 1.00
   d = (0.25x1.00) + (0.25x1.00)       = 0.50
```

(20) *Calculation of distance scores for* rondon no ofisu *and three examples*.

Based on the distance scores, (20a) is chosen as the best fit. The English side of the example is taken as the template, and the words *Tokyo* and *hotel* replaced by *London* and *office* (as given by the dictionary) to give *my office in London*. The system can also show the next best options, which in this case would be *the London office* and *a London office*.

There are a number of points that make the process slightly more complex however. The first is that for many words there are multiple translations: Japanese does not distinguish singular and plural, cf. *hoteru* in (15e,f). In the

[10] Strangely, in the original article, Sumita et al. multiply *sd* by *w*, which has the effect of *decreasing* the distance measure when there are varied patterns.

pattern `^AB`, the ^ symbol indicates adjectival form, so for example *nihon* may be *Japan* or *Japanese*. Thus, for the TL generation the translation patterns should really carry some grammatical information to guide the choice of surface form.

Another complication is that the form of the adnominal is also variable: in some examples the *no* particle is attached to a postposition, as in *kaiba madeno shatorubasu* 'shuttle bus to the conference site'. This is fairly easily handled by extending the distance calculation in (18) to include the adnominal.

The last example shows a case where the single Japanese word is rendered in English as a compound. Actually, this is not a big problem as long as the compound can be treated as a unit. More significantly, in the A *no* B construction in general the A and B can be noun *phrases*, not just simple nouns, as the examples in (21) illustrate.

(21) a. *50 nin hodo no guruupu* 'a group of about 50 members'
b. *sono kimpen no bijinesu hoteru* 'a business hotel in this neighbourhood'
c. *kaidan kikanchuu no sukejuuru* 'your schedule during the conference'

In this case, the distance measure has to be adapted to be able to compare single- and multi-word constructions. For example, the system should recognize that *kimpen no hoteru* is usefully similar to (21b).

A final (and realistic) difficulty is in handling inconsistency in the example set. In even our small example set, dates are translated in a number of different ways (e.g. *the 5th of August*, *March 8th*) and there are even examples of the same phrase with two different translations.

All of these are interesting problems which do give the student an insight into some of the pros and cons of EBMT.

7. Afterword

We have presented here a number of Prolog implementations of model systems. In fact, the author (and his students) have worked on one or two more, not reported here, including a French–English system of somewhat similar design to the Russian–English system described above, and (ironically, considering the history of Prolog – see above) a re-implementation of Météo. Students also had (at the time) limited access to Eurotra which, while not perhaps (intended as) a "toy" system certainly was written in Prolog!

References

Amores, J. Gabriel (2002) 'Teaching MT with Xepisteme', in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 63–68.

Bech, Annelise and Anders Nygaard (1988) 'The E-Framework: A Formalism for Natural Language Processing', in *Coling Budapest: Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, pages 36–39.

Bouillon, Pierrette and Katharina Boesefeldt (1991) 'Applying an experiemtnal MT system to a realistic problem', in *Machine Translation Summit III Proceedings*, Washington, DC, pages 45–49.

Chamberlain, Gillian T. (1994) *A model of an interactive machine translation system*. MSc Dissertation, Department of Language and Linguistics, UMIST.

Colmerauer, Alain and Philippe Roussel (1993) 'The Birth of Prolog', *SIGPLAN Notices* **28**.3, 37–52.

Fourman, Maximilian (1943) *Russian*, London: Teach Yourself Books.

Gal, Annie, Guy Lapalme, Patrick Saint-Dizier and Harold Somers (1991) *Prolog for Natural LanguageProcessing*, Chichester: John Wiley.

Hutchins, W. John and Harold L. Somers (1992) *An Introduction to Machine Translation*, London: Academic Press.

Kaplan, R.M. (1973) 'A general syntactic processor', in R. Rustin (ed.) *Natural Language* Processing, New York: Algorithmic Press, pages 193–241.

Kenny, Dorothy and Andy Way (2001) 'Teaching Machine Translation & Translation Technology: A Contrastive Study', in *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, pages 13–17.

McCord, Michael C. (1989) 'Design of LMT: A Prolog-Based Machine Translation System', *Computational Linguistics* **15**, 33–52.

Patterson, Rachel M. (1994) *A Prolog simulation of the example-based machine translation experiment of A.T.R. Laboratories*. MSc Dissertation, Department of Language and Linguistics, UMIST.

Pereira, Fernando C.N. and David H.D. Warren (1980) 'Definite Clause Grammars for natural language analysis – a survey of the formalism and a comparison with Augmented Transition Networks', *Artificial Intelligence* **13**, 231–278.

Sheremetyeva, Svetlana (2002) 'An MT Learning Environment for Computational linguistics Students', in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 79–87.

Somers, Harold (2001) 'Three Perspectives on MT in the Classroom', in *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, pages 25–29.

Somers, H.L. and R.L. Johnson (1979) 'PTOSYS: an interactive system for "understanding" texts using a dynamic strategy for creating and updating dictionary entries, in M. MacCafferty and K. Gray (eds) *The analysis of meaning: Informatics 5*, London: Aslib, pages 85–103.

Sumita, Eichiro, Hitoshi Iida and Hideo Kohyama (1990) 'Translating with Examples: A New Approach to Machine Translation', *The Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, Austin, Texas, pages 203–212.

Tomita, Masaru (1985) *An efficient context-free parsing algorithm for natural languages and its application*, PhD thesis, Carnegie Mellon University, Pittsburgh.

v. Hahn, Walther and Cristina Vertan (2002) 'Architectures of "toy" systems for teaching Machine Translation', in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 69–77.

Way, Andy (2002) 'Testing Students' Understanding of Complex Transfer', in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 53–61.

Whitelock, P.J., M. McGee Wood, B.J. Chandler, N. Holden and H.J. Horsfall (1986) 'Strategies for Interactive Machine Translation: the experience and implications of the UMIST Japanese project', in *11th International Conference on Computational Linguistics: Proceedings of Coling '86*, Bonn, pages 329–334.

Specification and Evaluation of Machine Translation Toy Systems - Criteria for laboratory assignments –

Cristina Vertan, Walther v. Hahn

University of Hamburg, Natural Language Systems Division

Hamburg, Germany
{cri,vhahn}@nats.informatik.uni-hamburg.de

Abstract

Implementation of machine translation "toy" systems is a good practical exercise especially for computer science students. Our aim in a series of courses on MT in 2002 was to make students familiar both with typical problems of Machine Translation in particular and natural language processing in general, as well as with software implementation. In order to simulate a software implementation process as realistic as possible, we introduced more than 20 evaluation criteria to be filled by the students when they evaluated their own products. The criteria go far beyond such "toy" systems, but they should demonstrate the students, what a real software evaluation means, and which are the particularities of Machine Translation Evaluation.

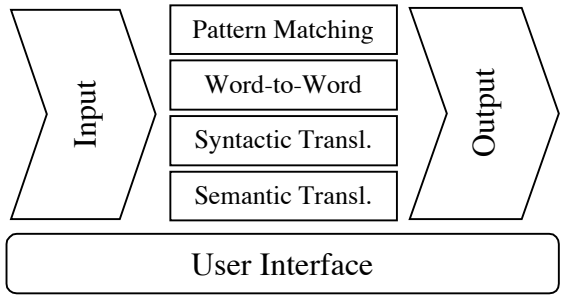
1 Introduction

Machine Translation (MT) is an important subfield both of Computational Linguistics and Natural Language Processing. Therefore academic education in MT addresses students in linguistics and computer science. Usually, according to the background of the students, courses are given separately to these two groups but with different methodologies: theoretical aspects and demonstration of tools for the linguists (Somers, 2001) on one hand, implementation of clearly defined algorithms for the computer science students on the other hand.

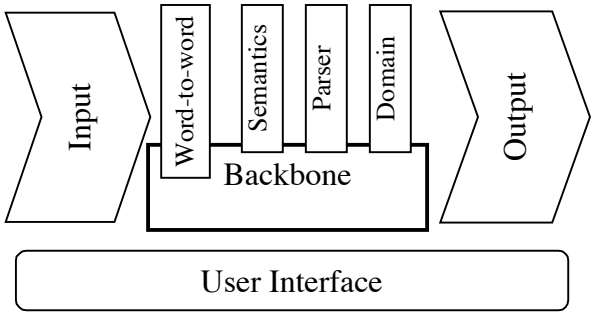
The alternative, to implement a realistic MT-system in one course is not feasible, due to the lack of time and missing background knowledge by the students. Very often they are facing the field for the first time.

A solution in between may be the implementation of one or several "toy systems", with rather limited language resources and limited functionality. In (v. Hahn and Vertan, 2002) the reader will find detailed examples of such toy systems, which have been developed mainly by courses for computer science students, but also including students from linguistics.

In one these courses the students had to implement (in small groups) parallel small systems based on either pattern matching, or word-to-word translation, or syntactic translation, or semantic translation. These sub-systems processed a corpus of app. 100 sentences, each, being controlled by a common user interface. The aim was to get a realistic idea of the possible contribution of each module in a real MT system.



In another course the students had to implement a classical centralised integrated system with a word-to-word pre-processor, syntactic and semantic modules, domain knowledge with an ontology and a user interface.



The aim of designing such systems is not only

- to offer to the students some interesting programming exercise, but
- *to make them conscious of what the implementation of a real system means.*

The remaining pages of this paper will explain details of this last topic of the courses, where we made the students reflect on questions like:

- "How will the system react with huge amounts of data or maximal throughput?",
- "How well are changes of the domain or changes of the languages supported?",
- "What about maintenance by users?"
- Or, more complicated: "What type of architecture is optimal for my required functionality?",
- "What kind of grammar is more suitable?"
- "How much from the original plans did we realise?"
- Which is the intuition of a possible user?"

In order to give the students the opportunity to reflect on all these types of questions, we asked the students, as integrated part of the laboratory assignment, to check 17 criteria for software quality evaluation (among them maintainability, system work-flow integration, or efficiency) and 7 criteria for the linguistic functionality, like lexical coverage, syntactic coverage or compatibility with (European) standards and formats.

The aim was to familiarise students with the evaluations of software projects. For some criteria, however, like maintanability or domain coverage, there was apparently no reasonable answer when working with a "toy" system, but the idea was to expose the students early enough to all aspects of software evaluation in general and machine translation in particular.

To make the whole implementation process more realistic, we also prepared criteria for the specification of the software, which had to be fulfilled before the implementation process. Both software evaluation and specification criteria follow general software engineering theory (Somerville, 1990)

In the following we will explain in detail each of the specification and evaluation criteria (both software specific and linguistic), and we will report about what the students learned from following such schemata.

2 Specification Criteria

a) Functional requirements: mainly the behaviour of input and output, both at system and module level

- On the system level the specification has to define the type and form of the input (e.g. speech or text, format of the input, e.g., file format and which other formats are supported, restrictions of text input from keyboard, menu-selection, audio formats, etc.) as well as the type and form of the output. To demonstrate the generality of the criteria we included such cases, where the output of a natural language processing system may be something else than a natural language utterance again.: A record from a databank or an action (in case of a robot system), or simply specific terms or web-links.
- The other part of the functional requirements concerns the module interfaces of the system. Specifying formats for input and output of each module right from the beginning, makes it much easier to work afterwards independently among the teams of this course. Each group can develop and test their modules with simulated input-data without waiting for completed work of the others.

b) Performance requirements

The students were asked to estimate which time behaviour has or is required for their system, and which resources it will need.

c) Usage requirements

Among students, this criterion usually is the most neglected parameter. They tend to assume that if input and output are in natural language, no special attention has to be paid to a user interface. A potential user, in their view, needs only a text field, where to type the input, and another part of the screen for the presentation of the output. We tried to raise their awareness for more specific requirements, esp. in MT systems: Dialogue windows for unknown words and errors in the input or the proper selection of labels and controls, facilities for reading in files, for pre-processing etc.

d) Embedding requirements

Under this heading the students are asked to specify, which hardware is needed for their system and which operating systems will be supported. A realistic scenario for the application has also to be discussed.

3 General Evaluation Criteria

The first group of criteria evaluates the usage of software, seen from the perspective of the user, under i) there follow criteria for the software product itself and under j) process quality critera

a) Adequacy

This point has to be assessed in reference to 2.c, i.e. how much from the specified user requirements are fulfilled, how user-friendly the system interface is, etc.. The students have to give precise examples, of situations where their system reacts adequate, and cases where improvements seem necessary.

b) Transparency

This evaluation criterion includes reasonable user estimations about processing errors, the plausibility of the system's behaviour in general or reasonable help facilities. Example: In a translation tool the user (ideally) has to be informed, whether a non-translated term is a word "out of dictionary", a proper name or simply an input error by the user. Of course at the level of toy systems we can not expect from the students (especially under time constraints) to tackle such problems, but they must be aware of their existence.

c) work-flow integration

As mentioned under paragraph 2.d, a possible scenario has to be specified initially for the system. In the evaluation statements the students are requested to explain to what extent their system would fit into an assumed work flow and with which additional time and costs their product can be adapted to other scenarios or work flow environments. Further, how flexible it is to functional extensions, because in a course such toy systems are designed exactly for the given or defined scenario. By including such a criterion we force the students to reflect about the difficulties of building a system, which is general enough to cover different scenarios and different work flow environments.

d) Specifications match

This requires a detailed comparison with all specification criteria, which of them are met by the implementation, what is still missing, and more, what is not conform with the specification criteria at all and why. The students must provide reasons why for example input and output formats were changed, or not supported, in the given form.

e) Reliability

The deterministic behaviour of the product, and its components has to be evaluated. As there is, e.g. no additional sensor input, translation systems must be deterministic.

f) Robustness

This is again an issue, where students have to learn a lot about real software behaviour and to make a very detailed evaluation. From our experience, they assume that their system works with all input data in the form that they require, and that the system runs in a "similar" way as with their test sentences. Their specifications usually cope only with the positive functionality "What to do" , not with functions to avoid certain behaviour. What happens, if the user forgets to specify parameters, if the user makes none of necessary actions, enters corrupted data etc. What to do about faulty input?

g) *Failure safety*

This criterion is mentioned only to familiarise the students with large scale evaluation procedures. For a prototype toy system it is not assumed that the implementation will include restart facilities, or that there are backup copies, but such aspects are important for real systems.

h) Efficiency

The efficiency of the program has to be estimated in terms of hardware requirements and consumption of resources, as well as the time required to perform certain operations.

i) Product quality

Under this title the students will correctly understand to briefly explain whether the program execution is correct, i.e. the expected behaviour is delivered. To refine the discussion we introduced the following sub-criteria:

- correctness (e.g. correct processing, complete correspondence to specifications)
- comprehensibility (e.g. structure of programs, choice of designators and names in the code)
- testability
- maintainability
- changeability
    - structural changes
    - functional changes
    - problem-type changes

The criteria mentioned so far are valid for any software product. In our case of toy translation tools, the problem of correctness is much more complicated due to the translation specific features. In contrast to classical software products where to any input a unique correct output must correspond, translation translation theory say clearly, that there

is more than one correct translation of the same input sentence. Moreover, the assessment "correct" for a translation is relative. For example, in the case of the Verbmobil system evaluation (Tessiore and v. Hahn, 2000), a lot of users were prepared to classify the output as "correct" already, if they could understand the meaning and pragmatics of the translation.

Usually the existent evaluation methodologies of machine translation, require the existence of a reference translation. A set of metrics are defined in the literature (Dabbadie and al., 2002) starting from the output and the reference translation. Our experience proved, that the existence of a reference translation can be even misleading for the students. For at least two of the toy systems, which were developed in our courses, we provided the students with a test corpus, consisting of about 70-100 sentences and their reference translations. At least three problems were encountered:

1. The students had a strong fixation on our reference translation: either they tried to tune their system artificially to deliver exactly the given reference, or they classified all translations as incorrect, which did not met perfectly the reference.
2. The construction of the (bilingual) lexicons is done strictly according to the reference translation: Only the morphology, meanings etc. encountered in the test corpus are included. As a consequence, the students did face the problem of disambiguation only in those cases, where we included it intentionally.
3. The development of the system was done strictly to cover the test corpus. Any additional sentence, would fail.

The scenario which we are applying now after this experience is rather different: At the beginning the students get no test corpus. Their first task together with the requirements in the specification task is to estimate what kind of sentences can occur in the given domain, and to design subsequently a lexicon which covers such situations. After the design phase and during evaluation we provide a test corpus, but only with sentences in the source language. This test corpus prevents the students from choosing only very simple cases, e.g., no anaphora and ellipses, no sub clauses or defective sentences

j. Process quality (e.g. quality of the implementation process, certification, quality of specification)

In contrast to the evaluation of the product, which addresses only the results delivered by the system and it's overall behaviour, process evaluation means the evaluation of the conditions, under which the software was produced. This covers the methodology for compiling the specifications, security measures, the design of tests, and the cooperation among the groups and with the customer. Here, the students have the opportunity to reflect about the quality of their production process and about the results of , e.g., underestimating time resources etc. Obviously, under time constraints, the code is not always documented, not always, explicit enough.

The aim of this "professional" software evaluation is not to over-criticize the results of the students but to show them what requirements are expected at a commercial level even for tasks, which are, by nature, not completely and formally defined and, by nature, vague, because this is the nature of language.

4. Criteria for Linguistic quality evaluation

In section 3 we presented evaluation criteria, which are valid for all software products. In the following we will concentrate on specific criteria for linguistic processing, in particular for translation tools.

a) Coverage

- Lexicon
- Syntax
- Semantics

As explained in section 2 .i), in a toy system the students will implement a reduced lexicon, a grammar which covers only part of the language and will deal only with restricted semantic problems. In our opinion, however, it is important that the student can define exactly the amount of linguistic features that they cover. Therefore they are asked to indicate:

- how many entries the lexicon has and to give examples of important word, which may occur in the given domain, but were not included,
- the annotations in the lexicon, the choice of lexicon type (stem lexicon versus full-form lexicon) and correspondingly, their morphological processing,
- types of sentences that can be processed, and types of realistic sentences which will fail,

- semantic phenomena, which are tackled and solved

b) Pragmatics

Here the students have to evaluate to which extent their software covers pragmatics aspects of the languages. Good examples are common directive speech acts like "The course is given in the city centre" (≠ Das Seminar wird in der Stadtmitte abgehalten", = in the university main building, not in CS building).

c) Compatibility

Translation tools make use of lots of resources (corpora, lexicons, grammars, etc.). Their development is time consuming, and therefore standardization efforts have been made since many years. The aim is to provide reusable resources. Therefore the students are asked to discuss, whether

- the format of their data, e.g., to what extent these meet existing standards and formalisms. If not, is the lexicon encoded in a reusable format (at least some XML version)?
- the grammar follows a well-known formalism (HPSG, functional grammar, etc.) and, on which basis the choice was done.

Concerning the languages, we usually define right from the beginning what is the source and what is the target language for the translation process. The students, however, must discuss if their program:

- can it be (easily) reversed to translate backwards: from target to source
- can it be adapted to new language pairs, and with which amount of work,. Here the general translation paradigm (transfer versus interlingua) can be addressed

Especially the linguistic evaluation can be a starting point for a broader discussion in the seminar about rather difficult issues in NLP:

- how much does a change of the lexicon design influences the design and the functionality of the whole system,
- is the lexicon part of the grammar (transition networks), then changes have influence on the whole grammar and the parser,
- how do technical ad-hoc decisions (easy implementation, time constraints, programming languages) restrict the whole system design and inhibit reasonable linguistic solutions.

Similar discussions can be triggered concerning the change of the domain. The change of the domain involves major re-implementations of at least the lexical resources and the pragmatic processes.

5 Conclusions

In this paper we presented criteria for the specification and evaluation of toy machine translation systems. to asses their quality The criteria can be grouped in two classes: general software evaluation criteria and specific linguistic ones. Both are used by the students to evaluate their own programmming. It is quite clear, that many of these criteria are by far too complex for such toy systems. The main aim is to familiarize computer science and linguistics students with real evaluation methodology. From our experience, the students had real difficulties to asses each point of the criteria list. However, at the end of the evaluation, they got some general ideas about why perhaps some of the methods, although locally successful, are not general enough, from which issues the success of an implementation depends and, last but not least, why the implementation of a machine translation system is not a trivial task.

6 Bibliographical References

Marianne Dabbadie and Anthony Hartley and Margeret King and Keith J. Miller and Mustafa El Hadi and Andrei Popescu-Belis and Florence Reeder and Michelle Vanni. 2002. A Hands-On Study of the Reliability and Coherence of Evaluation Metrics. In *Proceedings of the Workshop Machine Translation Evaluation – Human Evaluators meet Automated Metrics*, Third International Conference on Language Resources and Evaluation LREC 2002, pp. 8-16

Walther v. Hahn and Cristina Vertan. 2002. Architectures of "toy" systems for teaching Machine Translation. *In Proceedings of the 6th EAMT Workshop on "Teaching Machine Translation"*, Manchester, pp. 69-78.

Harald Somers. 2001. Three Perspectives on MT in the Classroom. *In Proceedings of the Workshop on Teaching Machine Translation* VIIIth MT Summit, Santiago de Compostella

Ian Somerville. 1990. *"Software Engineering"*, third edition. Addison-Wesley Publishing Company, Massachusetts

Lorenzo Tessiore and Walther v. Hahn. 2000. Functional Validation of a Machine Interpretation System:Verbmobil. In *Verbmobil: Foundations of Speech-to-speech Translation*, W. Wahlster ed., Springer Verlag, Berlin, pp. 611-634.

Teaching and Assessing Empirical Approaches to Machine Translation

Andy Way & Nano Gough
School of Computing,
Dublin City University,
Dublin, Ireland.

Email: {away,ngough}@computing.dcu.ie

Abstract

Empirical methods in Natural Language Processing (NLP) and Machine Translation (MT) have become mainstream in the research field. Accordingly, it is important that the tools and techniques in these paradigms be taught to potential future researchers and developers in University courses. While many dedicated courses on Statistical NLP can be found, there are few, if any courses on Empirical Approaches to MT. This paper presents the development and assessment of one such course as taught to final year undergraduates taking a degree in NLP.

1 Introduction

It is relatively uncontroversial to state that empirical methods in Natural Language Processing (NLP) and Machine Translation (MT) have become mainstream in the research field. Accordingly, it is important that the tools and techniques in these paradigms be taught to potential future researchers and developers in University courses. Many degree courses nowadays contain specific modules on statistical NLP (as it relates to word-sense disambiguation, parsing, generation etc.). While more and more courses on MT address the statistical approaches which are currently in vogue, it is more a case that this is done in passing in a couple of classes, rather than devoting a whole module to empirical approaches to MT. Indeed, in a trawl of the Web, we could find no such specific courses on statistical paradigms in MT. Of course, that is not to say that none exist; merely that none could be found by us.

Naturally, courses on MT need to take into account the students' skills and demands. Kenny & Way (2001) describe how MT is taught in one institution to students of differing backgrounds. A distinction is made in that paper between *users* versus *developers*: while language students and translators can be expected to be able to use translation tools in their careers as translators, students of NLP with a specialisation in MT might realistically be employed as designers and implementors of such tools in a programming or localisation environment.

This paper presents a course on empirical methods in MT taught to final year undergraduates taking a degree in NLP, focussing mostly on bitext alignment techniques and Example-Based MT (EBMT). These students have a strong background in programming, language skills and good competence levels in formal linguistics and NLP. Accordingly, the course is very practically oriented, and the students are expected by the end of the course to be in a position to develop a toy EBMT system. The presentation of the course in this paper is not intended to be prescriptive; rather, in reporting on the chosen methodology, it may serve as a basic model for others considering teaching such a module to similar students. In addition, by providing details of what worked and—more importantly, what did not—the hope is that others may benefit from lessons learnt.

The rest of the paper is organised as follows: in section 2, we discuss why teaching empirical approaches to MT is becoming ever more important, and provide some documentation of where such material is covered. Section 3 describes the content of the course taught by us. Section 4 describes the method of assessing the students to whom this course was delivered, and reports on the lessons

learned. Finally we conclude, and provide possible improvements and extensions to the course.

2 Teaching Empirical Approaches to MT

It is fair to say that empirical approaches to NLP and MT have matured to the extent that they may now be considered mainstream. Indeed, at any major conference on NLP/MT, nowadays one can expect to encounter more papers which favour an empirical approach than those which utilise rules and/or constraints.

If University students are to become familiar with such techniques and tools, dedicated modules have to be designed which address these topics. However, while there are many courses on statistical NLP already in existence, in a trawl of the Web, we could find no courses which solely address empirical approaches to MT. A similar parallel can be drawn with respect to textbooks: there are a number of volumes which specifically address statistical methods in NLP (e.g. Charniak, 1993; Manning & Schütze, 1999), but no textbooks are geared solely towards the concerns of empirical methods in MT. This hole in the market has recently been partially filled by (Carl & Way, 2003), which in turn may see an increase in the number of courses on EBMT, but there is still a need for a book on Statistical MT (SMT) itself (e.g. Brown *et al.*, 1990, 1992; Yamada & Knight, 2001; Soricut *et al.*, 2002).[1]

Nevertheless, while there would appear to be no courses (other than the one described in the next section) dedicated solely to the teaching of empirical methods in MT, SMT and EBMT have become so well established that any contemporary course on MT would be incomplete without at least equipping students with some superficial knowledge of these techniques. Some examples of courses which address these newer empirical approaches include, but are not limited to, the following:

- MSc. in MT, CALL and NLP at UMIST, UK;[2]
- MSc./Ph.D. Program in Language and Information Technologies at CMU, Pittsburgh.[3]

Others address the topic in modules on Empirical NLP, including:

- Programmes in Computer Science, ISI, CA;[4]
- Undergraduate Study in Computer Science at Brown University, Providence, RI;[5]
- Postgraduate programmes in Computer Science at UMIACS, MD.[6]

In addition, some of the newer textbooks on NLP/MT address these and related issues, e.g. Trujillo (1999, Chapter 8) goes into some detail on EBMT and SMT; Bowker (2002, Chapter 5) discusses the related area of Translation Memory (TM) systems; and to a lesser extent, Jurafsky & Martin (2001, Chapter 21) provide some detail on how empirical techniques can be used in MT. Melamed (2001) is geared specifically towards the exploitation of bitexts using empirical methods. However, until the advent of recent books dedicated to empirical methods in MT (e.g. Carl & Way, 2003), instructors in this area have had to rely on original papers and survey articles (e.g. Somers, 1999).

3 Course Content

Kenny & Way (2001) contrasts how courses on MT have to be tailored towards different sets of students with different backgrounds, even in the same institution. One of the authors of this paper also teaches a basic introduction to EBMT in two hours to a group of postgraduate students taking a degree in Translation Studies (TS). This is a small component of a module on Translation Technology. Given that TS students are more interested in TM tools, a superficial overview of EBMT, and especially the differences between TM and EBMT, suffices for this group of students. While both EBMT and TM require aligned corpora, for instance, the TS students

[1] While no standard University courses in SMT exist, Kevin Knight has given tutorials and short courses on SMT at various locations, including at the TMI in Kyoto in 1993, as well as at this MT Summit, of course! He also did a whole summer school at John Hopkins University in 1999, when he put together the *Statistical MT Workbook*, available at http://www.clsp.jhu.edu/ws99/projects/mt/mt-workbook.htm.

[2] http://www.ccl.umist.ac.uk/teaching/modules/3000/3003.htm, http://www.ccl.umist.ac.uk/teaching/modules/5000/5092.html

[3] http://www.lti.cs.cmu.edu/Courses/11-731-desc.html

[4] http://www.isi.edu/natural-language/people/cs562-2003.htm

[5] http://www.cs.brown.edu/courses/cs241/

[6] http://benreilly.umiacs.umd.edu/~hwa/cmsc828-02/

are far more likely to *use* built-in alignment tools such as Trados *WinAlign*, NLP students may be expected to *develop* their own alignment software.

The course presented here is geared specifically to a group of final year undergraduate students taking a degree in NLP. It would, therefore, be an inappropriate model for groups of students with differing backgrounds. The course in Empirical Approaches to MT taught by us consists of 3 hours a week lectures and a 2 hour practical session over a period of 8 weeks. The content of the course is as follows:

- Week 1:
  - Double lecture: Revision class on Perl.
  - Single lecture: Introduction to Course and Statistics-based MT.
- Week 2:
  - Lab: Perl exercises.
  - Double lecture: Corpus-based language and translation models. What corpora to use, and how to get them.
  - Single lecture: Alignment Methods: word-based, character-based etc.
- Week 3:
  - Lab: Sentential-level alignment using:
    - relative sentence position;
    - relative length of sentence.
  - Double lecture: Probabilistic Word (and Phrase) Models.
  - Single lecture: How to improve alignment with (a simple set of) heuristics (see Lab, Week 4).
- Week 4:
  - Lab: Add in cognates, paragraph markers, punctuation, HTML tags and other anchors to improve alignment.
  - Double lecture: EBMT. How it works, comparison with TM etc.
  - Single lecture: Marker Hypothesis as segmentation tool.
- Week 5:
  - Lab: Build word-alignment tool (i.e. probabilistic lexicon) using mutual information
  - Double lecture: Marker Hypothesis: advantages/disadvantages.
  - Single lecture: Problems for EBMT:
    - boundary definition;
    - boundary friction;
    - what examples to store etc.
- Week 6:
  - Lab: Build and Test EBMT system using 500 sentences of English, French and German data. Use sententially- and word-aligned databases.
  - Double lecture: Generalised Templates (cf. rules in rule-based methods in MT).
  - Single lecture: Generalised Templates.
- Week 7:
  - Lab: Improve model with generalised templates and word insertion.
  - Double lecture: Examples of EBMT systems.
  - Single lecture: Towards Hybrid Models.
- Week 8:
  - Lab: Finalise EBMT System.
  - Double lecture: Statistical MT.
  - Single lecture: Statistical MT.

The basic model followed here is that the material delivered in class during one week is put into practice in the labs during the following week. Essentially, the course is split into three chunks, namely Alignment (both sentential and word-level), EBMT and SMT.

In the introductory part of the course, students are made aware of the need for statistical language and translation models to be developed from large,

good quality, representative monolingual and bilingual corpora. By concocting toy corpora which do not fulfill these criteria, and asking students to calculate a number of unigram and bigram probabilities based on data contained in these corpora, it is quite easy to demonstrate that a number of undesirable effects follow when small, unrepresentative corpora are used. The advantages and disadvantages of bigram models are then presented to the students.

Despite the fact that the various mathematical techniques employed are, in principle at any rate, utilisable for any pair of languages, the fact that sententially aligned bilingual corpora exist only for a few language pairs renders these techniques somewhat less generally applicable. In order to try to overcome this problem, some consideration is given to using the Web as a corpus from which usable bitexts might be extracted (cf. Grefenstette, 1999; Resnik & Smith, 2003).

Some of the major algorithms for aligning bilingual corpora are then presented (Brown *et al.*, 1991; Gale & Church, 1993; Kay & Röscheisen, 1993). These are interestingly different, in that the method of Brown *et al.* uses a length-based metric which counts words, that of Gale & Church uses a character-based model, while Kay & Röscheisen require the use of a bilingual dictionary (a 'Word Alignment Table') constructed automatically from the bitext.

As for sub-sentential alignments, students are shown how to estimate co-occurrence using Mutual Information. With particular respect to EBMT, other methods of segmentation are also presented, especially Marker-Based segmentation (e.g. Veale & Way, 1997; Way & Gough, 2003). The 'Marker Hypothesis' (Green, 1979) is a universal psycholinguistic constraint which states that natural languages are 'marked' for complex syntactic structure at surface form by a closed set of specific lexemes and morphemes. Marker-Based MT constructs sets of marker words (e.g. determiners, conjunctions, quantifiers etc.) for source and target languages and segments a sententially aligned corpus into marker-headed chunks. ⟨source,target⟩ subsentential aligned pairs can then be automatically constructed if the number of marker chunks for a source string is equal to the number of marker chunks for the target equivalent, and the chunks are headed by marker words of the same category.

Students are also made aware of the need for extracted sub-sentential alignments to be made more general, in order to improve coverage and robustness. Some of the techniques presented include:

- Extracting transfer rules from examples (e.g. Furuse & Iida, 1992);
- Generalising by syntactic category (e.g. Kaji *et al*., 1992);
- Generalising by semantic features (e.g. Matsumoto & Kitamura, 1995);
- Generalising Parse Trees (e.g. Way, 2003);
- Generalising Strings (e.g. Cicekli & Güvenir, 2003; McTait, 2003);
- Generalising using Placeables (e.g. Brown, 1999).

These methods are embedded in a basic outline of the EBMT process. Comparisons are made with TM, some experience of which the students have had before. The major EBMT problems of boundary definition and boundary friction are presented: boundary definition is concerned with the problem that retrieved fragments may not be well-formed constituents, so that syntactic well-formedness may not be ensured in the generation of the target string, while the main problem with boundary friction is that context may not be taken into account in the retrieval process, so that ill-formed output is constructed despite the fact that the derivation of that output was correct according to the knowledge in the system's databases.

In addition, the storage of examples is discussed, issues pertaining to segmentation are put forward, and the matching and recombination stages of EBMT are explained. We then present the IBM Models 1 and 2 of SMT.[7] Finally, given that these students have previously taken a module on rule-based MT, we discuss possible hybrid models which combine elements of both paradigms as a more effective solution to the problems of translation.

[7]Note that some students choose to do their final year projects in this area, cf. http://www.redbrick.dcu.ie/~grover/project/index.html for a particularly good example.

4 Assessment

The course was designed to be an 'assessment only' module (i.e. no end of module exam), for two reasons. Firstly, being a second semester course, any final examinations would be scheduled during time which students would otherwise be spending on implementing their final year project (which constitutes 33% of their overall degree classification). Secondly, students at our University are classified only on their final year marks, so if students were to fail either the exam or assessment component of any final year module, they would not be eligible for an Honours degree. It was felt, therefore, that by having the module evaluated purely by continuous assessment, these issues could be best avoided.

There were two assignments:

1. a labtest on building an aligner (week 5);
2. a group presentation/demonstration on building an EBMT system (week 8).

The labtest was a 3-hour assessment, in which the students were individually asked to develop a number of programs in Perl, namely:[8]

- to calculate the average sentence length of the ⟨*source,target*⟩ sentences provided in terms of both words and characters;
- to calculate the ratio of ⟨*source,target*⟩ words and characters per sentence;
- to write a length-based sentence aligner, in terms of both words and characters;
- to compare the alignment results against a 'gold standard' provided;
- to segment the 'gold standard' reference solution according to the marker hypothesis;
- to propose sub-sentential alignments using the marker hypothesis.

In addition, there were three discussion questions on aspects of the course.

Note that all of these programs had been tackled in the lab sessions during the course. We considered three hours to be a reasonable time limit given that one of the authors was able to write programs to perform the various tasks in one hour. Nevertheless, we found that we had overestimated quite considerably the amount we thought the students were capable of in the time available: none of them completed all questions, and in general, too many students spent far too long on programs for which very few marks were awarded (as indicated on the question paper).

While the students' answers were marked benevolently (for instance, where students provided pseudocode instead of actual Perl code, full marks were given if the pseudocode was a complete solution to the problem at hand), using the original schema, over half of the class had failed, with the top mark being just 57%. However, a compromise was developed whereby the marks were divided by 0.7 in order to give a truer indication of each student's performance (top mark 81%, lowest 19%, average 53%).

The second assessment was a group project, where the students were divided into groups of three and were asked to develop an EBMT system, based on the sentence-, phrasal- and word-level alignments written in preparation for the first assignment. Marks were awarded both for system design and functionality, and for documentation. No one segmentation method was preferred over any other; indeed, some groups used the marker-based approach, others used a bigram approach, etc. The groups presented their systems to these authors, who found their efforts to be extremely good (highest mark 90%, lowest 57%).[9] Finally, in order to derive the final mark for the module, the first assignment was weighted 0.35, with the group assignment weighted 0.65. All students passed the module (highest mark 87%, lowest 44%, average 65%).

5 Conclusion

Empirical approaches to NLP and MT have reached a reasonable stage of maturity. It is important, therefore, that the tools and techniques underpinning these fields be taught to University students, who are

[8] For a set of freely available such tools, see Dan Melamed's page at http://www.cs.nyu.edu/~melamed/software.html.

[9] For a good example, consult http://www.computing.dcu.ie/~sfoy-cl4/ebmt.html.

likely to form a pool from which future researchers and developers in these areas are to be found. While dedicated courses on statistical NLP have made their way into many University curricula, we were unable to find any courses on empirical methods in MT in a trawl of the Web. Similarly, while a number of textbooks have appeared on statistical NLP, the first such book on one of the flavours of empirical MT, namely EBMT, has only recently appeared.

This paper presents the development and assessment of one such course as taught to final year undergraduates taking a degree in NLP. It focusses mainly on Alignment, EBMT and SMT. It was designed so that student performance was evaluated purely in terms of continuous assessment. We commented on the problems that arose in scheduling a laboratory test of the students' understanding of alignment, both at the sentential and sub-sentential levels. Nonetheless, when asked to develop an EBMT system in small groups, the students rose to the challenge and performed well on this task.

As for improvements/changes to the course, the students may be assessed on a more ongoing basis in weekly labs rather than in one laboratory examination. In addition, the material in (Carl & Way, 2003) may be used in study classes, with students presenting this material to the class on a weekly basis. Finally, we may choose to focus more on the building of SMT systems, using the excellent, freely available Egypt toolkit.[10]

In sum, despite some teething problems, it can be said with some confidence that the module was successful. The developers and teachers of this course have certainly learned from the experience, and it is hoped that others who are considering the development of similar courses may find some value in the sharing of our experiences. That said, nothing in this paper is intended to be prescriptive: if the course structure and methods of assessment are of use to others, then fine, but if some other model is chosen, then we too would hope to benefit from the development of similar or related material.

References

[1] Lynne Bowker. 2002. *Computer-Aided Translation Technology*, University of Ottawa Press, Ottawa, Canada.

[2] Peter Brown, John Cocke, Stephen Della Pietra, Fred Jelinek, Vincent Della Pietra, John Lafferty, Robert Mercer and Paul Roossin. 1990. A Statistical Approach to Machine Translation, *Computational Linguistics* **16**:79–85.

[3] Peter Brown, Jennifer Lai and Robert 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.

[4] Peter Brown, Stephen Della Pietra, Vincent Della Pietra, John Lafferty and Robert Mercer. 1992. Analysis, Statistical Transfer, and Synthesis in Machine Translation. In *4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, Canada, pp.83–100.

[5] Ralf Brown. 1999. Adding Linguistic Knowledge to a Lexical Example-based Translation System. In *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 99)*, Chester, England, pp.22–32.

[6] Michael Carl and Andy Way, eds. 2003. *Recent Advances in Example-Based Machine Translation*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

[7] Eugene Charniak. 1993. *Statistical Language Learning*, MIT Press, Cambridge, MA.

[8] Ilyas Cicekli and H.Altay Güvenir. 2003. Learning Translation Templates from Bilingual Translation Examples. In *Recent Advances in Example-Based Machine Translation*, M. Carl & A. Way, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, pp.255–286.

[9] Osamu Furuse and Hitoshi Iida. 1992. An Example-Based Method for Transfer-Driven Machine Translation. In *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation: Empiricist vs. Rationalist Methods in MT (TMI-92)*, Montréal, Canada, pp.139–150.

[10] William A. Gale and Ken W. Church. 1993. A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics* **19**(1):75–102.

[11] Thomas R.G. Green. 1979. The Necessity of Syntax Markers. Two experiments with artificial languages. *Journal of Verbal Learning and Behavior* **18**:481–496.

[12] Greg Grefenstette. 1999. The World Wide Web as a Resource for Example-Based Machine Translation Tasks. In *Proceedings of the ASLIB Conference on Translating and the Computer* **21**, London, [pages not numbered].

[13] Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing*, Prentice Hall, Upper Saddle River, NJ.

[14] Hiroyuki Kaji, Yuuko Kida and Yuji Morimoto. 1992. Learning Translation Templates from Bilingual Text. In *COLING-90, Papers Presented to the 13th International Conference on Computational Linguistics*, Helsinki, Finland, pp.672–678.

[15] Martin Kay and Martin Röscheisen. 1993. Text-translation alignment. *Computational Linguistics* **19**(1):121–142.

[10] http://www.clsp.jhu.edu/ws99/projects/mt/

[16] Dorothy Kenny and Andy Way. 2001. Teaching Machine Translation & Translation Technology: A Contrastive Study. In *Proceedings of the Workshop on Teaching Machine Translation, MT Summit VIII*, M. Forcada, J-A. Pérez-Ortiz & D. Lewis, eds., Santiago de Compostela, Spain, pp.13–17.

[17] Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA.

[18] Yuki Matsumoto and Mihoko Kitamura. 1995. Acquisition of Translation Rules from Parallel Corpora. In *Recent Advances in Natural Language Processing: Selected Papers from the Conference*, R. Mitkov and N. Nicolov, eds., John Benjamins, Amsterdam, pp.405–416.

[19] Kevin McTait. 2003. Translation Patterns, Linguistic Knowledge and Complexity in EBMT. In *Recent Advances in Example-Based Machine Translation*, M. Carl & A. Way, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, pp.307–338.

[20] Dan Melamed. 2001. *Empirical Methods for Exploiting Parallel Texts*, MIT Press, Cambridge, MA.

[21] Philip Resnik and Noah Smith. 2003. The Web as a Parallel Corpus. *Computational Linguistics* **29**(3) (to appear).

[22] Harold Somers. 1999. Review Article: Example-based Machine Translation. *Machine Translation* **14**(2):113–157 (revised and updated in M. Carl & A. Way, eds. 2003. *Recent Advances in Example-Based Machine Translation*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp.3–57).

[23] Radu Soricut, Kevin Knight and Daniel Marcu. 2002. Using a Large Monolingual Corpus to Improve Translation Accuracy. In *Machine Translation: From Research to Real Users, 5th Conference of the Association for Machine Translation in the Americas (AMTA-2002)*, S. Richardson, ed., LNAI 2499, Springer Verlag, Berlin/Heidelberg, Germany, pp.155–164.

[24] Arturo Trujillo. 1999. *Translation Engines: Techniques for Machine Translation*, Springer, London.

[25] Tony Veale and Andy Way. 1997. *Gaijin*: A Bootstrapping, Template-Driven Approach to Example-Based Machine Translation. In *International Conference, Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria, pp.239–244.

[26] Andy Way. 2003. Translating with Examples: The LFG-DOT Models of Translation. In *Recent Advances in Example-Based Machine Translation*, M. Carl & A. Way, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, pp.443–472.

[27] Andy Way and Nano Gough. 2003. *wEBMT*: Developing and Validating an Example-Based Machine Translation System using the World Wide Web. *Computational Linguistics* **29**(3) (to appear).

[28] Kenji Yamada and Kevin Knight. 2001. A Syntax-Based Statistical Translation Model. In *39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computational Linguistics*, Toulouse, France, pp.523–530.