MARGARET KING
Université de Genève, Institut pour les Études Sémantiques et Cognitives

EUROTRA - A European System for Machine Translation

1. Lessons from the past

Previous articles in this journal will have given the reader an idea of the state of the art in currently operational machine translation systems This article describes a system which is planned, and which it is hoped will be developed by all the Member States of the European Community acting together, within the framework of a single collaborative project.

The motivation for such a project is manifold. First, we have learnt a great deal from the systems which already exist, both in terms of what to do and in terms of what not to do. To take the positive lessons first: the most important, of course, is that machine aided translation is feasible. This lesson is extremely important. After the disappointments of the 60's, it took a great deal of courage to persist in the belief that it was worthwhile working on machine translation. A great debt is owed to those who did persist, whether they continued to develop commercial systems with the tools then available or whether they carried on with the research needed to provide a sound basis for more advanced systems. Mad it not been for their stubbornness, machine translation would now be one of those good ideas which somebody once had, but which proved in the end impractical - like a perpetual motion machine, for example -instead of being a discipline undergoing a period of renaissance and new growth.

Secondly, we have learnt that problems which once seemed intractable are not really so. Looking at a book on machine translation written in the early 60's the other day, I was surprised to find the treatment of idioms and of semi-fixed phrases being discussed as a difficult theoretical problem. Of course, idioms still must be treated, and must be treated with care, but operational systems have shown us that they can be successfully translated. This does not mean that no system will ever again translate "out of sight, out of mind" as "invisible idiot", but if it does so, it will be for lack of relevant data, not because mechanisms to deal with such phrases are not adequate.

It would be possible to make a fairly extensive list of similar problems, which once gave machine translators nightmares but now only cause mild insomnia. Suffice it to say that experience with existing systems has given us the knowledge that such problems can be solved, and the courage to find ever better ways of solving them.

At a technical level, too, we have learnt a lot from existing systems. Early, not very successful, machine translation systems were dictionary based, essentially taking one word at a time and trying to find its equivalent in the target language. As a fairly natural reaction to the disappointing results obtained by such a method, there was something of a swing later to concentrating on the linguistic analysis parts of the system, those parts which tried to determine the underlying structure of the input text in order to translate at a "deeper" level. Practical experience has taught us that even though analysis is crucial, dictionaries retain a great importance, in that any working system will rely heavily on large dictionaries, sometimes containing whole expressions as single entries, rich in static linguistic information on each entry and serving as essential data for the translation process. So we have learnt to pay attention both to the initial design and coding of dictionaries, and to their manipulation in terms of large data bases which must be constantly updated and maintained.

Based on rather more negative experience, we have learnt that system design is all important in a machine translation system. This can be said rather differently, by saying that we have discovered that a translation system is necessarily going to be *big* - and that big systems need special treatment. No one person, or even group of persons, can hope to keep a large computer program under control if it is written as an amorphous mass. It will be impossible, when things go wrong, as they inevitably do, to find out where in the program they went wrong, or why It will be impossible for an outsider who has inherited the program from its original author(s), to understand what they did or why they did it. So a large program must be made as modular as possible: that means that it must be broken up into well-defined sections, each one with its task clearly known, together with the starting information it will work on and the results it can be expected to give. In addition, it must be well documented. It should be written in a computer language as easily readable and comprehensible as possible, and should be provided with an abundance of commentary explaining its function.

None of the above paragraph is specific to machine translation systems: indeed, its content is by now the received wisdom passed on even in elementary courses in computer programming. But one aspect of systems design is particular to machine translation, and that is the absolute necessity of a rigid distinction between algorithms and data. This distinction, although it sounds esoteric, is in fact familiar to anyone who has ever followed a recipe. In their standard form, recipes give first a list of materials required and then a set of instructions saying what to do with these materials. The list of materials corresponds more or less to the data, the list of instructions to an algorithm. In the case of analysing language, the data will consist of, for example, dictionary information and a description of syntactic or semantic grammar rules, whilst the algorithmic part of the system consists of instructions about how to apply the rules and the dictionary information to a text in order to determine its structure. There is a constant temptation to mix up the two: to include inside a dictionary entry, for example, a little instruction to go and look for a particular dictionary entry following this one, or to put into the algorithm trying to find noun groups the information that adjectives come (sometimes) between articles and nouns. The consequences of falling into this temptation can be very nasty indeed. The most common consequence is that it becomes impossible, eventually, to change the system in order to correct mistakes or to enlarge the range of texts it can deal with. A relatively minor change, meant to deal with one specific linguistic feature, may affect the treatment of other features in quite unforeseeable ways. So information and what to do with it should be kept apart.

2 New ideas in EUROTRA

What has been said in the previous section applies to any machine translation system to be designed today. In this section, let us look at some of the aspects of EUROTRA which are specific to it.

The most obvious of these is, of course, its multilinguality. Traditionally, translation systems are constructed for specific pairs of languages, and take advantages of any similarities between the language pair. If extension to a new language pair is required, normally some or all of the system must be rewritten. EUROTRA is the first system which is designed from its conception as a multi-lingual system. Initially it will deal with the six languages of the European Community, and further languages may be added later. This is achieved by keeping the analysis and generation of each language independent of all the other languages, so that the module which carries out, say the analysis of Danish, is the same whether the system is translating from Danish into Italian, from Danish into English or any of the other possible target languages.

Of course, there is still a part of the system which depends on the specific language pair involved: this part, the bridge between the two languages, is the transfer module. One of the aims of the system design is to keep the transfer module as small as possible, precisely limiting it, wherever feasible, to finding equivalences between lexical units of the source language and lexical units of the target language. This may sound dangerously like word-to-word translation, which, its is well known, gives disastrously bad results. But in fact, since the choice of lexical unit may be made on the basis of very complex analysis of the underlying syntactic and semantic structure of the text, we are a long way from word-to-word-translation. It should not be forgotten either that a lexical unit may be considerably more than a single word: an idiomatic expression or a technical term composed of several words, for example, may appear as a single lexical unit Nor should it be forgotten that generation of the translation is not, in the case of EUROTRA, limited to generating the correct morphological forms and trying to gel the word order right. Quite complex manipulations of the structure of the text may be involved. Sometimes these are required for stylistic reasons. Often though, they are required in order to create the *correct* structure (not only the stylistically appropriate structure) in the target language. The quick rule of thumb is: If it can be done within the grammar of the target language without access to the source language lexical unit, do it in generation. A good illustration is the translation of "ought" into French, where the corresponding lexical unit in French would in some circumstances be "devoir", in others "falloir". The transfer module will decide which to choose, but the generation module will be capable of generating the appropriate structure (finite verb + infinitive for "devoir", impersonal + que + subjunctive for "falloir") without knowing whether the original text was in English, in Dutch or whatever. Thus, as can easily be imagined, the generation phase is very powerful, and can do a great deal to improve the intelligibility and the readability of the translation.

Multi-linguality affects many other aspects of the system too. If the analysis phase is independent of the target language, this automatically means that the analysis modules must provide an analysis adequate for the transfer into any one of the target languages. In linguistic terms, this means providing a 'deep' analysis of the logico-semantic structure of the text. The importance of this can most easily be seen by considering prepositions. French 'par' has different translations into English in each of the following sentences, where the correct translation is determined by the logico-semantic relationship between the prepositional phrase and the main predicate.

On y arrive par I'escalier = The place is reached by a stair.
II regarde par la fenetre = He is looking out of the window.
II court par Ies rues - He runs about the streets.
Par latitude 10o nord = In latitude 10° north.
Venez par ici = Come this way.
Par ou a-t-il passe? = Which way did he go?
Par un jour d'hiver . . . = On a winter's day . . .
Par le froid qu'il fait = In this cold weather.
II a ete puni par son frere = He was punished by his brother.
Accable par l'inquietude = Overcome with anxiety.
Je I'ai appris par les Smith = I heard of it from the Smiths.
II a reussi par I'intrigue = He succeeded through intrigue.
Elle est une dame remarquable par sa beaute = She is a woman remarkable for her beauty.
II le fait trois fois par jour = He does it three times a day. (These examples are taken from Harrap's Shorter French-English dictionary, and by no means all of the entries under 'par' are repealed here.)

Similar lists could be drawn up between all the language pairs to be dealt with, showing, finally, that there is no straightforward mapping between the use of prepositions in any single language pair.

If all that is handed to the transfer phase is the simple fact that there is a prepositional phrase in the original, the transfer phase must determine the logico-semantic relationship between that phrase and the rest of the sentence in order to choose the correct English preposition - and so must every other transfer phase. Six languages gives a total of thirty language pairs and therefore thirty transfer modules. So the work of determining the logico-semantic relationships must be done thirty times, if it is done in transfer. It is clearly simple common sense to do it in the analysis phase, once for each language Thus multi-linguality forces analysis to go consistently beyond the syntactic level to produce an analysis of the logico-semantic structure of the text adequate for any of the target languages, where a bi-lingual system could take advantage of any accidental overlaps between its two languages by only going beyond a superficial analysis when it proved necessary.

This should not be interpreted to mean that *all* disambiguation should be done in analysis, including disambiguation of single lexical items where the structure is not affected. To do this would mean that a word which was ambiguous in only one of the languages would have to be treated as ambiguous in all, a strategy which would prove problematic in two ways.

First, an enormous amount of contrastive study would have to be done in order to know when a word was ambiguous, and the ability to distinguish the different senses somehow communicated to those building each separate analysis module. To see how difficult this might be, it is only necessary to think of classic cases like "know" in English being equivalent to "connaitre" and "savoir" in French. English children do not think of "know" as ambiguous (it isn't) and often have difficulty learning to use "connaitre" and "savoir" correctly.

Secondly, the analysis modules for each language would become very large. Imagine that some word has two senses in only one of our languages. Then five analysis modules out of six are forced to distinguish two senses where really only one exists, and in twenty transfer modules out of thirty, the lexical unit chosen will be the unit which covers both senses, i.e. the word will be "re-ambigualed". Clearly there is a trade-off here, where the important trick is to avoid unnecessary work whilst at the same time ensuring that necessary work is done in the most economic way possible.

If, as the criterion for what disambiguation should be done where, we take that the underlying structure of the text should be unambiguous at entry to transfer, whilst individual lexical units not affecting the underlying structure may be left ambiguous until the transfer phase, then the depth of analysis aimed at is still very ambitious, and requires the use of very powerful semantic tools never before incorporated into a full scale system.

Multi-linguality affects the treatment of dictionaries as well In order to obtain maximum benefit from considering the dictionaries as large data bases, they are to be organised into a single multilingual lexical data base. This will make it possible to provide many very useful tools to help in the dictionary makers' work. For exam-ple, the addition of an entry for one language can trigger an automatic check of whether transfer entries exist into the other Ian-guages, with consequent automatic signalling of any deficiencies Similarly, it will be possible to check for conflicts between dictionary entries.

Another novel aspect of EUROTRA is implied in what has already been said: it is designed to be an extensible system. Extensibility here means not only the possibility of adding new language pairs without being forced to re-write what already exists, but also means being able to incorporate into the system the results of new research work as they appear. The importance of this is evident when the history of research in computer treatment of language is considered. The last fifteen years have seen a revolution in the study of linguistics which has had immediate effects on computational linguistics.

A great deal of current research, especially perhaps in the field of Artificial Intelligence, can be expected to produce results which, when applied in a working system, would enable us to tackle problems which for the moment have to be left unresolved. Certain problems of pronoun reference, for example, require a use of common sense knowledge of the world so extensive as to be impracticable with techniques currently available. Precisely these types of problems obsess workers in Artificial Intelligence: they may well, in five or ten years, have found practicable solutions.

So it makes sense to plan a system in such a way that it can profit from continuing research, and is not limited to the frozen state of the art at the time the system was designed.

The modular design of EUROTRA allows for the future. A new way of tackling analysis or generation can be experimented with without the rest of the system being affected. New language pairs and new subject areas can be added without the rest of the system being rewritten.

Even the grammars used in analysis, transfer and generation are modular. They can be divided up into sub-grammars dealing with some specific aspect (say, the analysis of noun groups or the establishing of one particular semantic relation) which may be as large or as small as the grammar writer wishes. This gives a great freedom which allows the grammar writer to experiment with new techniques

A combination of modularity and the rigid distinction between algorithms and data also allows grammars or the control packages which apply the linguistic data to the text to be exchanged between groups. One group may develop a particular technique and pass it on to a second group who wants to try it out as a black box, without the second group needing to know any details of how the insides of the black box work - in the same way as one buys a television set - thus giving an extra possibility of experimentation with new techniques and at the same time avoiding unnecessary reduplication of work.

But if all this is so, it is clear that something must hold the system together: it is all very well for the muscles to be supple, but they must be attached to the skeleton if they are to work at all. In our case, the skeleton is what in computing terms is called the "software", the basic programs around which the system is built. It must be useable on different computers - some of them perhaps not yet developed - and, besides providing the linguist who writes the grammar and the dictionary coder who writes the dictionary entries with a comfortable and convenient language to write in, it must also provide powerful and easily useable tools to allow the programmers or those working on the linguistic parts of the system to follow through the action of the programs and to find out, when necessary, what goes wrong and why. (It is almost impossible to follow the detailed action of a large computer program without the aid of the computer itself: there is simply too much detail for a human brain to attend to all at once)

But the software must also try to make life easy for the end users of the system: the people who will actually use it to do translations, as opposed to the people working on the development of the system

One important way to accomplish this is to provide the end-user with very up-to-date text-processing tools. Imagine that a reviser is revising a translation produced by the system: it would be pleasant if he could work directly on a terminal with a video-screen, with the computer itself taking care of producing a clean copy - and even of layout details like where the end of a line should come or automatic re-numbering of pages. It should be possible, too, to change every occurrence of a word or phrase throughout the text by signalling the change just once, or to move large stretches of text around by a simple command.

The end-user might not always want the same quality of output: for the sake of having a quick translation - and machine translation can be very quick indeed - he might be prepared to accept a lower quality. EUROTRA will allow him to do this, by allowing him to combine the modules (remembering that grammars loo are modular) in any way he wants, in order, say, to suppress the more detailed levels of analysis.

3. Why EUROTRA is feasible

Section 2 seems to describe a very ambitious project: everything about it is on the grand scale. Why do we believe it can be done? Part of the answer lies in the recent history of machine translation in Europe. Whilst in America, those who continued to work on machine translation after the collapse of the 60's concentrated on the production of operational systems, in Europe the groups who continued concentrated much more on pilot projects intended to develop and test new research results. Thus there exist in Europe a number of relatively small groups who over the last ten or fifteen years have been carrying out the research needed to establish the theoretical basis of advanced machine translation systems. This high level of expertise is there, waiting to be used.

But it cannot be used to full profit without the scattered pockets of expertise being brought together in a single project. Naturally enough, groups have concentrated on different aspects of the translation process. So there are groups who have developed very powerful techniques for dealing with morphology, with syntactic analysis, with semantics. But no single group is expert in everything. By bringing them all together, all can benefit from the experience of others and the co-operative enterprise of producing the first ever large scale multi-lingual system can be based on a sum of knowledge which is greater than the individual parts.

It is here that the European Community plays a crucial role. Only under the auspices of an entity like the CEE is it possible to envisage ' the organisation and development of a collaborative project involving large numbers of people working in groups in all the countries of the Community.

4. Organisation

This brings us to the question of how the project is to be organised. It is planned that a team in each country will be responsible for the production of analysis and generation modules for its own language. (Where two countries share a language, The team will be a joint team.) Teams made up from the two groups involved will be responsible for the production of the transfer modules between each language pair. To avoid producing an incoherent monster of a system, a separate team, independent of the different language teams, will be responsible for ensuring communication between the different groups, making sure that the modules will in fact fit together via a well-defined interface structure and for producing and maintaining the basic software. This way of organising the project is only made possible because of the modularity of the system.

It is expected that the first version of the operational system will be ready within about five years from the start of the project. This first version will still be fairly limited. Its grammars cannot be expected to deal with all possible sentence structures and all possible problems of ambiguity, but it is nonetheless expected to produce translation of distinguishably higher quality than any currently operational system, and the design of the system guarantees that it can constantly be improved to push up the quality even further. Similarly, this first system will probably only deal with one subject area, but the system design ensures that it will be easy to add others.

5. Conclusion

This article has tried to give a very rough idea of a machine aided translation system, EUROTRA, currently being planned under the auspices of the Commission of the European Communities. The project as described is ambitious, both in terms of scale of the initial system and in its claims to extensibility. But no translator should fear finding himself without work as a consequence of the development of this or similar systems. There will always be lexis which cannot and should not be translated by machine: a classic example is any text whose ambiguities are deliberate. A machine cannot have the translator's sensitivity, which allows him to know when an ambiguity should be carefully preserved if he is to remain faithful to the original. What this system is intended to do is to remove the burden of banal and boring work from the translator, thus simultaneously leaving him free to concentrate on enjoyable work, and solving the ever-increasing problem of finding enough translators - especially in the more unusual language pairs - to go round.