[From: P.Whitelock et al. (eds.) Linguistic theory & computer applications (London: Academic Press, 1987)]

Translation

Rod Johnson

I have more or less the same preamble as Graeme [Ritchie] did yesterday. The original intention, as I understood it, was that there would be a number of 'theme sessions' about various kinds of applications. As it turns out MT is the only application left on this list of themes, which makes one feel a little bit exposed. Nonetheless I would want to maintain that under current circumstances, MT is worth spending a bit of time on, because it raises some quite interesting theoretical methodological questions which haven't really been raised up to now, or only touched upon, and in view of this I'll try to concentrate on that particular collection of issues as far as I can.

Just a few clarifications: There are all sorts of things that come under the rubric of MT systems. There are an awful lot of products on the market these days (that most of you probably know about) which are really just hyped-up word-processors or some kind of on-line dictionaries, and even some things which one might think of as MT systems proper finish up, when you examine them in some detail, looking like big sentence dictionaries. These are not what I really want to talk about. I don't think they have any particular immediate relevance to the general set of topics we're supposed to be addressing. The characteristics of the sort of MT that I think is interesting, and for which the topics we've been discussing these last few days may be of some relevance, are something like the following:

(i) I'm going to assume that MT systems are big. That doesn't here mean 'big' so much in the sense of wide coverage of language (which I'll come to in a bit), but rather 'big' in the sense that, in order to make one, you have to appeal to the knowledge of quite a large number of people and you have to do so in a fairly extensive way. The reasons for this are that MT systems - of the sort I want to talk about - involve a number of languages (cf. point (iii) below), and you're unlikely to get the sort of language expertise that you want from a single individual or even two or three individuals in one place. This fact has some consequences for the organisation of MT systems, the way you set them up and the way you design them, which I think impinge on the kind of theoretical notions you have to have before you start.

(ii) Similarly I'm assuming that we're talking about, programs which are intended to translate texts from a fairly unrestricted source, with some constraints perhaps on the text domain, but not with any artificial limitations on some set of texts of which you have prior knowledge and beyond which you're not really expecting to go.

(iii) As I said before, things get interesting once you get beyond two languages. If you stay within two languages you can cheat in all sorts of ways without it being really apparent.

(iv) I'm assuming that these programs don't involve human intervention. You don't have the opportunity of appealing to human intuitions about things that you can't make the program handle. This is an unrealistic assumption in general, but it's not a bad starting assumption to see where it gets you before you give in and decide that there are some things that you can't do. It's a good thing to decide a posteriori, in the light of some sensible theorizing of what you're not going to be able to do, rather than just to state beforehand and then find some way of doing the things that are left.

(v) Lastly I would like to restrict the sorts of MT programs being talked about to ones which are in some strong sense linguistic. What I suggest by that is that it is reasonable to propose that the construction of a mapping between texts which are chunks of language is primarily organized according to linguistic principles. This is not to exclude the famous real world knowledge, which is always a bone of contention in this area, but simply to say that the driving force here is some kind of linguistic theory, or some set of linguistic beliefs. You may graft onto that some more information which is extra-linguistic, but you don't start off by saying, we need lots of external knowledge, and by the way what extra bits of information do we need about language in order to get the job done? The assumption being made here about MT being linguistic is quite the converse of what some people in some areas, even people who've claimed they're doing MT, overtly say they're doing. And so I think it needs to be stated that that's the assumption underlying this discussion.

Here are some pragmatic requirements that come from outside that are likely also to have an effect on the way that you put the things I've just mentioned together. These are probably not exclusive to MT by any means, but they do turn out to be very important in view of some of the other characteristics of MT systems.

(i) In this domain in particular, you want the systems that you make to be easy to extend. Of all applications for language processing, MT programs of the kind characterized above are going to be particularly open-ended - they're never going to be finished in any reasonable way. The amount you have to change them may decrease with time, but you will never get everything right, you will never be able to predict every single phenomenon that's going to come up in your set of texts. Even if the only open-endedness finishes up being in the lexicon, it will still be the case that there are new things you have to know about. Given the circumstances of these conditions, you are not really allowed to suppose that if you don't have the linguistic apparatus to treat some input text available then you throw the text out, and if the same phenomenon occurs a couple of weeks later then you will throw it out again. Your responsibility, if you get something wrong, is to try and put it right. So it's not as if you can stay within some restricted language. In addition to that, it's quite normal in this area to have to add a new language: to take what you've got and to be asked to make it handle some completely new language in the same way as you're handling the existing ones. And it also has been the case in the past, although rather less successfully implemented, that people have been asked to handle new text types using the original apparatus.

(ii) You want the system to be modular – everybody says this, but there's a reasonable amount of truth in it here I think – because of the size of the thing and because of the way that you expect to put things together, quite apart from any engineering considerations about making it easy to build and to maintain. You really do want to be able to take parts of your system and plug them together in different ways. This is partly the 'more than 2 languages' question. If you're handling three or four languages, it really is very sensible, perhaps almost a priori necessary if you are going to do this thing rationally, to take something that handles, say, French and use at least some part of that intact for doing translations into German or into Italian or Arabic or whatever, and not to have to rebuild the whole thing all the way up.

(iii) The last requirement is 'transparency', which has had a mention from time to time. The idea here is that, as for any big program in fact, if you come back and look at the system after a few weeks or a few months or if some new people come along and look at what someone else has written, you really do want to be able to understand, or you want the new people to be able to understand, what the thing is supposed to be doing. This is particularly true because these programs, if you ever get them running at all, tend not to be ephemeral. Once a system is there, and some sponsor or other has paid a lot of money for it, then they expect to use it over a long period of time, and this long period of time is such that you cannot normally anticipate that the same people will be working on it at the beginning as finish up working on it at the end when the owner actually decides to throw it away and get something else. So it becomes important, if only for engineering reasons, that you can understand the code. The thing is a 'program', with some linguists around who are its 'programmers', and they have got to know what is going on in there if the thing is to be maintained in any sensible way. You can't rely on there being one person somewhere who wrote it who understands what it does.

Perhaps these are all truisms; I think they probably are. They're the sort of statements that are made about large programs in any domain. Why I think it's still important to make these statements, even if they're very familiar to you, is that it isn't habitual that we think of language handling programs as large programs. An awful lot of them are small programs, and even set out to be small programs, and the engineering principles are perhaps different if you try to make big programs like this properly.

Now a few parenthetic remarks here: there is still a tendency among MT people, which many of you will have observed, to get defensive about doing MT, because there's a feeling abroad that MT is a disreputable activity. And especially the kind of MT that I tried to characterize earlier on. I'm not going to spend very long on this particular sociological issue. I certainly don't want to raise the standard arguments of feasibility which people in MT these days usually raise in defence of what they're doing. I do however, in this forum, think it's worth making a few points about the interest of doing MT as a piece of research, something which is very rarely said by people in this field. Here are some reasons why it strikes me (and also lots of other people who work in this area and who make the same comments) why this kind of MT is worth doing in its own right:

(i) First of all, as a piece of computational linguistics, it is very difficult to imagine any other computational linguistic endeavour within which you are forced to deal with several languages at once. You don't just deal with them because there is some interesting phenomenon which someone tells you about so you think you'll look at it. You have to deal with significant subsets of a number of a different languages simultaneously. Not just that, but you have to give an account of each of them which is compatible with the accounts of all the others, because in the end you're talking about some kind of similarities between languages which you want to capture in order to get the translation done. MT forces you to do that, and that's very good mental hygiene in this business.

(ii) The second thing that this kind of MT makes you do is look at rather large subsets of these languages. You can't get away with looking at two or three phenomena and building some program that treats them and not worrying about the other things because they're getting in your way. If you're going to treat the stuff at all then you've got to treat more or less all of it, and again in a fairly consistent way.

Now I know these things are obvious. The reason I'm restating them is because it seems to me that, while they may be a nuisance, and while they may make the task very hard, they also justify the task in a very strong sense, in that it's salutary to look at language in such a way that you cannot ignore certain things, and to look at languages particularly.

(iii) Another thing that doing this makes you conscious of is the value of notation, another thing which computational linguistics has tended in the past to disregard as not being all that important, essentially because the people who have done the computation are the people who have had the linguistics in their heads. This is changing, but if you think back over the sixties and seventies then notational issues have not been considered important. Phrases like 'notational variant', or 'these systems are equivalent' spring to mind. The point here is that because of the size of the user community, because of the large number of people who have to put their ideas together and make them fit, notational issues, subjective; and aesthetic as they may be, become as crucial here in whatever linguistic theory you adopt as they are in programming languages. And we all know how certain notational issues in programming languages have been quite significant over the last few years.

(iv) It also makes you think about the lexicon. Lexicons for this kind of activity have to be very large. It's very difficult to restrict them artificially, and even if you start off by restricting them you soon find that, if you want to make the thing open-ended, then it is going to grow very fast, even in restricted domains. There are a few exceptions to this. The famous one is Meteo (Chandioux 1976), which everyone cites, Meteo being the MT system that works because the people who made it very fortunately came across a domain that really suited the purpose absolutely. But there aren't many of those domains left.

(v) My last comment is a bit cryptic. I'm throwing it in at the end as a kind of after-thought because it's not had very much mention in here and it's one of those small but quite important points that perhaps deserves talking about. What you realize if you perform this exercise is that you have to have, as I said, a large description of the language. It's not clear that the description of the language that you are prepared to accept is going to be the same as the description of the language you are prepared to regurgitate at the other end. Some trivial instances of this: the one I always cite particularly is the so-called split infinitive in English. Now this is the kind of area where it may well be that someone who wants to buy an MT system off you, or someone who asks you to build it, will simply refuse to accept output texts which split infinitives. I know this is a trivial notion – I think one can find better examples with a little bit of thought. I keep it simple to make a point, the point being that if you therefore have a linguistic description which does not contain certain phenomena because you require it in order to generate your texts, you are going to get in serious trouble as soon as you try to analyze texts in that same language. I think that this particular phenomenon extends a long way beyond the split infinitive. Things like the order of certain clauses in Italian. The fact that a clause with siccome is required by some speakers of Italian who are very fussy always to come before its matrix verb, although you find lots of texts where siccome comes after it. Another example is the distribution of inanimate subjects with non-passive verbs in Japanese, which will probably come up again. I am given to understand that large numbers of Japanese speakers, and perhaps particularly those who will be managers and people requiring translations, will be very angry if they start getting texts with non-animate subjects of a certain class of action verbs, whereas I also understand reliably that these things turn up in texts quite regularly, and you can't wish them away from texts.

The idea here is that you may in the worst case need completely different descriptions of your analysis language and your synthesis language. Of course you don't want to do that. You would like some commonality between them, if possible. How do you get it? Is it possible to take a description and decompose it in such a way that you can use some parts of your English grammar twice (and not now in the 'modularity' sense, of using a description of English for translations into different languages): sometimes in analysis grammar and sometimes in synthesis grammar, substituting other parts? I don't know if anyone has asked these questions but I've never heard them talked about.

So – going back – I want to claim that some at least of these questions should be interesting even for people who think MT as an engineering practice a waste of time, and I want to claim that MT is one of the application, perhaps one of the few applications, that actually makes you think of these questions in the first place. End of apology.

Back to the real point. Here's the standard characterization of translations that we want to use: we want to see translation as an equivalence relation between classes of source language and target language texts. Jan [Landsbergen] called these equivalence classes – or these pairs of equivalence classes, perhaps – 'possible translations' of each other. That's the relation we want. Here are the standard perspectives these days on how you get it. The first two are perhaps fairly standard; the last one is fairly recent and rather less canvassed than the others, but we have a principal representative of it here, so 1 couldn't leave it out. The three general ideas on how you actually compute this relation are as follows:

(i) The first, perhaps the standard way, or the one you encounter most frequently, is the contrastive way. This decomposes the relation in such a way that you define abstract representations for both your source language and your target language, and you have the principle that these representations canonically represent the classes of translation-equivalent texts. Translation then comes down to mapping between those canonical representations of texts, and that's essentially contrastive.

(ii) This way of seeing the relation contrasts with the interlingual view. This view these days is particularly upheld by people from Yale (Carbonell et al 1978; Lytinen & Schank 1982) and their associates. It tends to be the way that non-linguists see translation, I think. By non-linguists I mean people who do not have good familiarity with more than one language. The idea is that there is a class of canonical representations which fully characterize all the information necessary to induce these equivalence classes of possible translations in all languages that you are dealing with. Objects within that representation theory are called 'interlingual objects' and the representation theory is called 'interlingual theory'. You can translate between source language text and this neutral representation, and you have another translation between objects in neutral representation and target language texts. and that is translation.

(iii) The third view, which is typically characterized by Jan Landsbergen (this volume), is the view that the translation relation is captured in terms of derivational equivalence. You have descriptions of your source language and your target language, and instead of pairing representations of text in source and target language, you pair items of the descriptions. You analyse your source language, and instead of aiming at some representation, you keep the trace of the analysis, and the translation equivalence between languages is captured in terms of pairings between items in that trace. I'm not being specific about Jan, I'm trying to characterize the general idea. I won't pursue that now – I want to say overtly that, as perhaps already implied, the view that I can best characterize is the first one, the contrastive one. That's the one which I feel I can talk about, although that need not inhibit discussion on the others.

Here's an interesting note about all of these things. The notion that you have an abstract representation which is somehow canonical is taken by some to suggest that that representation necessarily is the meaning representation, and that the job therefore involves to some extent mapping between texts and their meanings. I just want to point out that, although this may in practice turn out to be so (although you'll probably find that the grain that you want is much finer than the grain of simple meaning representation), in theory what you're after is translation equivalence. Now this may be meaning equivalence; but translation equivalence is not judged by meaning equivalence monolingually. Translation equivalence in the end is judged by observations about pairs of texts, and it's judged on the basis of the texts. So as long as you can capture canonically the formal essence of these classes of texts through some appropriate pair of representation theories, it doesn't actually matter if the thing turns out to be a meaning representation. In particular, even if you have to appeal to meanings in order to perform this computation, that doesn't necessarily require that the representations which you translate in the contrastive view be meaning representations. Similarly, it may well be that you have to undo some quantifier scopes in order to get your translations right. In order to undo these quantifier scopes you need some kind of predicate logic representation. But just because there is something that requires you to make that representation in order to do some disambiguation, say, it does not therefore follow that the basis of your translation should be a predicate calculus representation.

This is worth saying.

Here's the real point now. With those givens, what do we look for in a linguistic theory that will support this kind of activity? And there's a rider on that. Some people in MT, unfortunately – and perhaps this is one of the reasons for its bad reputation – would rather ask the question 'Why do we need a linguistic theory at all?' I hope that question is not going to be raised here. I would rather phrase it as: given that it's clear that we need a linguistic theory to support this activity, then what should it be like? Here are some notions which again are fairly straightforward.

(i) I want to claim that such a theory should be declarative, i.e. that descriptions in such a theory should be declarative. So although MT is clearly a procedure, and indeed you can pick out sequential components of this procedure, I want to maintain that you would really like the descriptions that drive that procedure to be declarative, and this is motivated not just by aesthetics but by the sort of engineering considerations I was trying to talk about earlier. You want to extend the system and perhaps extend it in a fairly massive way in the end. You don't want to fiddle around changing pieces of program, and there are many cases where if you did have a piece of imperative program procedure, you would find the need from time to time not just to add new routines, but actually to get inside this piece of imperative programming and change bits of the sequence of the procedure and things like that. That's exactly what you want to avoid, with all these people getting in there and tampering with it. I guess the typical analogy in conventional computational linguistics to something like an ATN managed by, say, 10 or 15 people all contributing to it. I think you want to avoid the mess that that's likely to give you, and I think that suggests that you want some kind of declarative notation as your programming language, and you want the theory to supply that.

(ii) There's a question of 'conceptual versus computational modularity'. We don't have time to go into that, but some people will know what I mean: that the decomposition of what you want to say is likely to be different in terms of computational organization from the way it is in terms of the organization that you have in your head when you think of its linguistics. A declarative notation should help you to maintain that distinction and to make it not matter to the people who are dealing with each of those two cases.

(iii) You want it to be 'effective' in the technical sense that we have used a couple of times. You want to use an effective procedure, or at least you want a procedure, and if it's not effective then you would like to have some kind of grasp on the places where it isn't. This I know is obvious, but again there are some people in MT, the extremists of the linguistic MT point of view, who are almost ready to claim that this doesn't matter very much, that the linguistic description is what counts. Of course you need a meld of both things.

(iv) Because of what I was saying before about canonicity and the translation relation, I want to claim that your linguistic theory should give you a sub-theory of canonical representation of some kind. Now it may not give you all the substantive pieces of that theory, but it should at least give you the formal apparatus within which to develop it.

(v) And then the last two points – there are good engineering and conceptual considerations for wanting your theory to be restrictive. There are large numbers of people around, you want the system to be modular, so you don't want people going away and writing any kind of description and then finding you can't fit them together. So you want lots of constraints which guide the way that people actually write their descriptions down. At the same time, because of this open-endedness, you don't want to be so restrictive that, if it goes wrong, there's nothing you can do about it except go back and rebuild the whole thing, because there's no way you can capture within the constraints you've got whatever this new phenomenon is that's causing the thing to go wrong. It seems to me that there's a tension here. There are two conflicting sets of demands, and there's no real answer to this. But certainly there are some theories that give you a better compromise between those two things than others, and 'better' here is a completely aesthetic judgement, or perhaps empirical one day.

Those are the sorts of considerations involved. Now here's a short summary of the issues that people might want to talk about that seem to be interesting from various points of view. The first one says: 'Is the best way to model translation as something that passes via representations or something that passes via derivations?'

If it goes through representations – or indeed derivations, only it's a little bit harder to say then – do you want your representation, in theory, to be interlingual, that is totally universal, or do you want to be contrastive? And remember that 'totally universal' means you can't even have idiosyncratic lexical items any more. You have to decompose your lexical items so that they're language-neutral as well. That's a big enterprise, especially when you've got several hundred thousand of them. And completely decompose them, not just decompose some parts to help you to do some disambiguation, as Wilks (1973) does. Take them completely to pieces.

And then the question of analyzing and synthesizing: do you want the language descriptions that drive the analysis and synthesis to be the same? If not, what's the relationship between them? What do you expect from a linguistic theory to help you to capture this relationship, whatever it is? Take the notion of flexibility, which Stu [Shieber], particularly, has been advocating. He says something like: you want your theory to give you lots of escape hatches so that you don't get stuck when you're doing developmental work. And compare that to the constraints that you need if you are going to get a lot of people in a lot of different places describing a lot of different languages all saying roughly the same kinds of things, and you want to enforce that compatibility.

Then there's the issue of the incorporation of non-linguistic knowledge, which I have avoided, not because it's embarrassing (though perhaps it is), but because this gathering is primarily to talk about the linguistic applications. But if people think this is an important issue then there's certainly no reason why it shouldn't be raised. And then the last point, which I think is the crucial one: are there now linguistic theories around which are adequate to support at least some part of this MT enterprise? If they're not adequate, then can we conceive how we might extend them to do the job? If they don't exist at all, then why not? Is it that the demands of this kind of work are so different from the demands of all other computational theoretical linguistics that there's no apparatus around? I don't think that can be true. And there we are.

Pete WHITELOCK: I'd like to start the first discussion point, and illustrate it with a problem that we have in our English-Japanese system (Whitelock et al 1986). I'd like to argue against the feasibility of doing translation by derivation. Let me give you an example: as most of you will know, passive constructions occur widely in many languages. Certainly in English and Japanese there's something that looks very similar in the sense that the active and passive forms are inflexionally distinguished, they're often truth conditionally equivalent (modulo quantifier scope), and they involve demotion of something that's considered the subject to some oblique role. Now on the surface of it that would make it look as though English passives and Japanese passives were going to be translation pairs but in fact (I don't think this is crucial but just to talk about it in some framework) there is a parameter of Universal Grammar I think it's called 'PRO-drop' (Chomsky, 1982:28 et passim) - which is set in Japanese but not in English. Now because you can have subjects in Japanese which are just not present, what that means is that the textual distributions of passives in English and Japanese are totally different. Alternatively, if you're thinking of handling this sort of thing by relating derivations, it means that the translation rules for pairs of rules that deal with the short passive and the long passive in English are in fact going to be totally different. The long passive as it occurs in English is primarily used as a mechanism for topicalization, and you have a different mechanism for topicalization in Japanese. So you wouldn't want to translate it as a passive. Whereas with a short passive you might very well want to do so. So in fact you're saying in English short passives and long passives are syntactically very different sorts of things, but they're not really.

Jan LANDSBERGEN: I'd like to react to that. I do not fully understand the example, but I think that the point you want to make is that our approach with isomorphic derivations will have the effect that the grammar of the source language may be influenced by the grammar of the target language. In general that is true. What I hope and expect is that it will not influence it too much. In addition it should be noted that syntactic notions like passive and active are not necessarily expressed explicitly in the derivation trees. The kind of information that is retained during translation may be a more abstract notion, for instance what the topic of the sentence is. In one language a sentence may be in the passive voice, while in the other language the corresponding main verb does not allow the passive voice, but there may be another mechanism to get an NP to the front, of the sentence. That would be sufficient. So in a derivational approach one may have to translate via somewhat more abstract notions than just active and passive.

WHITELOCK: OK, but it does seem though that what you're going to end up with is a pair of linguistic descriptions that's actually a single one and there's no way that you can actually separate out very much and say this is knowledge of English, this is knowledge of Dutch or whatever. You really have to motivate the description in terms of the relation between the two languages and you have no single language information.

LANDSBERGEN: It is single language information, but its form may be influenced a bit by the other language. As I said yesterday, grammars are artefacts. There are many ways of describing a language. The particular instance we choose out of the set of possible grammars will be influenced by other languages.

Mary McGee WOOD: I wonder how far there's a tension between the different obvious practical reasons for doing this tailoring of the source language grammar to help you to get to the target language. I'm thinking of Rod's criterion that one should be able to handle more than two languages, and perhaps you ought to talk even about language families (because it's a bit of a cheat to handle, say, English and Dutch and then say 'Look, I can go easily to German', or Italian, Portuguese and then Spanish). But it's different if you have a system which does English to German and then it can go to Japanese easily as well. I think there may be a tension – in principle one wants to be able to extend, but in practice one wants to have grammars that will work together easily and keep your system working.

Henry THOMPSON: My intuition on the basis of not having, heard this approach before was that it was applicable in direct proportion to the typological similarity of the languages, and that is as you move to language pairs or language triples that cut across major typological boundaries, that it seems less plausible. Now maybe that's a naive perception but I think that Peter was trying obliquely to refer to that same problem.

JOHNSON: If you believe that this is implausible – and I must say it certainly sounds it to me too – then to what extent is this a reflection of the Montague (1973) grammar that Jan claims motivates the whole approach, and to what extent does it reflect on the approach which in some way does not depend on the particular grammatical theory?

THOMPSON: I think that as an intuition coming from somebody outside the tip of the arrow on this, what I'm saying is not worth very much, but it is I think an obvious danger.

LANDSBERGEN: The question at issue here seems to be: suppose that we have a system for Dutch, English and Spanish, what happens if we want to add Japanese? I never claimed that this would be possible. I think it might be possible to add German, French, etc., because they are related to the other languages. But if we would want to translate into Japanese, we would have to take it into account right from the start. If the distance between two languages is as large as it seems to be between English and Japanese (but I do not know a word of Japanese), then it may be wise to work only on this language pair.

May I say something in favour of the derivational approach? The danger of translating via deeper levels of representation is that one does not really solve the translation problem, but splits it up into a number of translation problems that are not necessarily easier to get hold of. The deeper these levels of representation are, the more difficult it may be to understand what you are doing at these levels.

WHITELOCK: What the transfer-through-representation approach does is actually give you a place to localize the fifteen percent of hacking that you've got to do at the end. It's unavoidable. It's linguistic hacking in the sense that there's no real reason why the parameter settings for a given pair of unrelated languages should be anything other than arbitrary, so in fact what you have to do to do transfer is essentially linguistically hack.

LANDSBERGEN: If you want to hack, why don't you allow us to add a few extra rules to the English grammar to suit the Japanese?

WHITELOCK: But then your English grammar is polluted: it's no longer motivated by considerations of English only.

Doug ARNOLD: I think there are two issues being conflated in this discussion. I think in fact Pete's example, though interesting and instructive, conflates the two. One is the essentially representational issue of what the basis of the comparison or contrast between English and Japanese is - it's clearly not to do with active and passive. And the problem then is devising a representation which captures the true basis of the difference, which is this complex of focus and the free deletion of pronouns in Japanese which you're not allowed to do in English. But the other issue is this whole question of the derivational approach to translation, and what your [Landsbergen's] question, indeed your system, requires is that these two things go together. You have a single representation, a single path of analysis, and you translate off the derivation of that analysis. Now it is quite possible to combine that with the transfer approach, where transfer goes between two derivation trees, but those; derivation trees aren't the derivation trees for the surface structures of the language: they're for some other representation. For example, a representation of your own invention which described the focus and the presence of superficially absent pronouns and all that wonderful stuff.

WHITELOCK: There's no reason why you would want to map anything other than the deep representations. Why should the surface syntactic representations be relevant at all?

ARNOLD: Well normally, for simple kinds of grammar, the derivation trees and ordinary representations are identical, and the interesting case is where you have an ambiguous language. So suppose your semantic representation language is ambiguous: then the question is, do you want to have to parse that in order to work out the correct translation, or do you want to just look at it and know what the correct translation is? Well, if you just have to parse it, it looks more efficient (though this isn't the best argument) to do it off the derivation tree, which avoids the problem of having to parse the representation.

JOHNSON: Don't you choose a semantic representation language which is non-ambiguous?

ARNOLD: It's very difficult to do that, unless you're prepared to be interlingual about it, because 'ambiguity' in this case means 'ambiguous with respect to the target language'.

JOHNSON: Is anybody prepared to address the last of those points [existence of adequate linguistic theories for MT], which I think is the really important one?

KAPLAN: Let me ask a prior question. I of course believe that in order to do machine translation well, you have to have a deep theory of language, and really understand what you're doing, but that's because I'm not a banker. If I was bank-rolling this kind of activity, what I would want was something that I could go out and sell, that worked well enough. Would I really invest in a linguistic theory at all? What I want to know is the answer to this question, so that when people ask me this question, I'll know how to answer it. So you tell me. Why isn't machine translation like medical diagnosis? All these expert systems now are doing medical diagnosis, and they don't understand about disease. They use 'if...then' kinds of pattern-action rules, and it doesn't really matter how diseases get there, or how the genes go bad, or whatever. What matters is that if you see this kind of symptom you do this kind of thing. And usually that works, and if it doesn't then you try the next thing. So why don't we use expert system technology?

THOMPSON: That's like saying 'What's the matter with sentence dictionaries?'.

KAPLAN: All I'm saying is, what's the matter with buying yourself a knowledge engineer, or a knowledge engineering work-station, and getting an expert translator, and sitting down and capturing all his rules of thumb about ...

WHITELOCK: That is the way to do it.

Jackie KNIGHT: I'd like to dispute the first point that that's the way medical expert systems work. The ones that people want to use don't work like that. That's exactly the reason why they haven't been taken up.

KAPLAN: What about oil-well logging systems, or any number of other systems?

KNIGHT: But the move is to go to the underlying knowledge, because people want that, they are able to use it. They need explanation, they need to respect the system. In order to do that it has to explain its own reasoning. But that's if you have to go deeper. So if you produced a machine translation like that, people may well not want that anyway.

Graeme RITCHIE: There's another point which is narrowness of coverage. These expert systems work on tiny domains. I'm sure these guys round the table here could knock together a really excellent MT translation of persona1 pronouns between English and German without any problem.

SPARCK JONES: But Ron's point is not that we need expert systems, but what is the knowledge that's supposed to be captured within the expert system? The guy may be a pretty good doctor, but it doesn't follow that he has a theory of healing, or a theory of the working of the human body. He hasn't got a theory. So the question is, what do we mean by 'theory'? He may have a lot of knowledge, and you want to capture the deeper knowledge of the doctor, and translators may have a lot of deep knowledge – if you see the way some of these translators work, it's perfectly obvious they've got a lot of deep knowledge. It doesn't follow that they have a theory of language.

Lieven JASPAERT: There's two things you should remember. Graeme made a good point, and that is the size of the domain. The domain of translation is very big and many expert systems work on a much more restricted domain. The second point is that in the case of translation, the domain is very badly understood. We know that translation exists, we know that translators can do it. We know they have lots of knowledge about translation. But it's very difficult to characterize first how a human translates, and it's even more difficult to characterize how a machine should translate. If you want to solve the MT problem, you must be able to get a grip on what translation is, and you don't do that, I think, by giving the people who work in the MT field a very liberal theory, an environment in which they can do anything they want. It's nice for experimentation, but they will never understand where they go wrong, where they go off the tracks, because everything is possible. I think that makes a case for a very strong linguistic theory that forces you in certain directions, maybe not the directions you want, but at least at some point you will be able to see that this is not what you want. Here is where your theory makes you do things that you don't want to do. The case for a strong theory and a notation that mirrors the strength of that theory is what is to be investigated.

THOMPSON: This is basically the software engineering argument. I think it's a good argument, but perhaps there was some suggestion, it seemed to me, in Rod's presentation that he was hoping that there was some other argument as well as the software engineering argument. Now it seems to me that the software engineering argument is a very good one. It's not clear that you need a better one, and most of the standard software engineering arguments go through more or less metaphorically – I think actually not terribly metaphorically at all, you can take them just about straight. That's I think what you were offering. It seems to me that Ron and perhaps Rod were looking for something else as well, some arguments from some different perspectives. Is that fair?

JOHNSON: I guess so.

Nick OSTLER: Is there any really authoritative statement of what constitutes an adequate translation? For example we had a dispute yesterday as to whether something should be called 'paraphrase' or 'real translation', and the force of Peter's objection was to say that your grammatical theories are talking about active and passive, but in actually assessing whether a translation is adequate or not you should be talking about focus and anaphora really. So it seems to me that what you really need in order to answer your last question 'Do adequate linguistic theories exist?' is an authoritative statement from people who do translations presumably, or people who consume translations, as to what is actually required in terms of the relation, because then you'll be able to see to what extent linguistic theories are addressing the right problems at all. It seems to me that linguists are concerned that abstract constructions present in one language should be present in the translation. Consumers of translations may be much more concerned about the overall effect of the document. Of course that is a rather ill-defined notion.

THOMPSON: The problem is that that of course varies widely with the domain of the application. To get to the extreme case, if you take the Finnish legal system, where the law is written in Finnish and Swedish, then there's very clearly a strong and strenuous definition of what constitutes an acceptable translation, and the operating version of it is that it shouldn't be to anybody's particular advantage to be tried under one text as opposed to the other. And that's a pretty serious business: one of things you go to a lawyer for in Finland is to ask him which language you should bo tried under. That's extremely strenuous. On the other hand the classic story is that for people who are interested in what's happening in Russian physics and who are physicists, almost anything counts as a translation as long as it gets the lexical relations right about *80%* of the time. That's if you believe the publicity. I don't know if that's true or not.

SPARCK JONES: Jonathan Slocum put it last year (Slocum 1984:558). He said it's an adequate translation if people will pay money for it. Now that offers us no leverage on linguistic theory at all!

OSTLER: Could we not proceed inductively then, taking these authoritative statements, perhaps relativizing them to their contexts, and get some general principles out of that? Because otherwise the whole thing is so ill-defined, I don't see how you are ever going to answer that question 'Do adequate linguistic theories exist?'.

ARNOLD: The answer to the question would surely be a linguistic theory of MT, or a linguistic theory of translation, because the characterization that would come out of whatever inductive procedure you set up will be just a linguistic theory of MT, wouldn't it?

OSTLER: No, I feel it would be a theory of MT; it wouldn't be a linguistic theory of MT unless you realized that the things that people were really worried about were all linguistic things.

JOHNSON: Texts are linguistic objects in an obvious sense.

OSTLER: Yes, but the aspects of them that linguists are worried about may not be the ones that concern consumers of translations. The active-passive one was a good concrete example I thought. There are enough of these institutional environments where they have these requirements for translations, whether in the Finnish legal system or in the Canadian broadcasting code, or whatever. Perhaps there might be scope for an actual survey of that sort of thing. It might give some surprising results - I don't know.

JOHNSON: Perhaps. Was it Henry who just said that there were so many domains and so many text types around ...?

THOMPSON: What people are looking for when they're looking for a translation varies tremendously.

SPARCK JONES: It's not the text or the domain but the uses that people want to make of it.

OSTLER: But what are those uses? You see we don't know. We know it varies, but we don't know concretely what it is we have to worry about.

JASPAERT: Well I used to teach in a translators' school for a while, and it struck me that even those people who did translation weren't able to characterize or have a theory about how it is in human translation. Then you're asking the same question for machine translation, and I think it's impossible. You have to do it the other way round. Adequate translation is the kind of translation that, given a certain theory and a certain approach to doing machine translation, then if people are willing to pay money for it, then you have adequate translation. So I'd be a bit bottom-up about it actually.

JOHNSON: I don't know if you could get away with that.

ARNOLD: There's an assumption behind the discussion of a little earlier. It comes out clearly in Henry's remark that the software [engineering] argument *is* quite relevant and you don't need to look anywhere else for the motivation for looking for theoretical principles. I think that you would not have made that remark about any of the other things that have been discussed - you would not say that about morphological theory or syntactic theory. The software argument for doing it this way is enough.

THOMPSON: Au contraire! I think what Rod and Mitch [Marcus] were saying last night about morphology goes very much along those lines. Rod was slightly apologetic by the time he got to the most extreme version of it, but it was still very much that version. It was saying, maybe what's true about the relevance of morphology and lexical look-up is that it gives you software engineering-style leverage. Now on the syntactic front, yes I would admit to other motivations .

ARNOLD: I wonder, are there other motivations for MT? It seems to me there are. It seems to me that translation is a perfectly respectable thing to be interested in. And the most intellectually respectable way to be interested in it, in this day and age is a computational one. It surprises me that that isn't more widely touted as a view of the business.

KAPLAN: Well, the point that I was trying to make was that there are different purposes and different evaluations that you might give. You might really be interested in machine translation as a scientific question and really want to develop a theory of that as an important natural phenomenon. You might want to develop a theory of it, and that has its own justification and formulation. But there's also this commercial aspect to it, and there might be quite different criteria.

WHITELOCK: I don't know: your characterization of how to achieve commercial MT actually sounds like a very good research programme to give some data as a basis to build a linguistic theory of translation, because I don't think anybody knows how to start. To actually say, well you produce some sort of representation and then an expert translator has to understand the primitives of that representation system, and then says 'If this and this and this, then this is the output'.

THOMPSON: This is crazy. This is like saying that it's perfectly sensible to start out by trying to have a theory of the mechanics of perception involved in three-ball cascade juggling before you have a theory about grasping. If you want to take the task of translation as an interesting human ability which you think it's relevant to develop a theory of, then surely it's insane to start out and tackle that as an undecomposed problem. There's a greater dependency on the simpler problem of language comprehension, for example.

WHITELOCK: I was not saying that. I was assuming that you had in the first instance some sort of off-the-shelf theory which builds you some sort of representation which abstracts away from the surface string, that was all. And I would think that the existing theories that we've got are more or less adequate in that respect.

KAPLAN: Commercially you get this translator in there and you give him a text and you say 'Here's a sentence. How would you translate that sentence and why?'. And he says 'Well, there are some big words at the beginning and some little words at the end, so I do it this way'. You know, maybe if you're doing genetic engineering experiments you need 500 [rules], but maybe for machine translation it needs 2000. You keep doing it.

Bob MOORE: The thing about translation is that the knowledge that expert translators have that's specifically knowledge about translation is based on fluent understanding of the languages that are translated. So it would not make any sense to attempt the expert systems-type approach until you'd got the technological base that gives you the equivalent of fluent understanding of both languages, and that's several lifetimes' work.

WHITELOCK: And of the relation between the two.

KAPLAN: That's what I question. Whether to get a practical useful acceptable machine translation system you really have to have all that knowledge.

WHITELOCK: Yes, I think so. There's a sort of assumption, which came through what Henry said, that all you have to do to do translation is to understand the source language and generate the target language.

THOMPSON: That's a minimal preliminary requisite, not a sufficient condition.

WHITELOCK: But that is patently insufficient.

ARNOLD: The relevant comparison with a fully automatic MT system, the kind of thing that Rod was talking about, isn't an expert system. Expert systems are semi-automatic, because the output of an expert system requires interpretation: you have to put it together. I think you could easily build an expert system kind of thing for doing translation – indeed they exist. It's something like a big lexicon, with a few bits of sentence dictionary thrown in. What you get out at the end is a thing which is bits of source text, and then you give people some clever editing facilities. I think you have expert systems that do translation, but they don't do fully-automatic translation.

SPARCK JONES: In connection with what Ron and Henry were saying, you have different reasons for doing MT. Now a reason we haven't talked about very much is that, supposing you have a linguistic theory, and you're actually a person who thinks that one of the ways to really see whether your linguistic theory actually stands up is to make it work for something computationally, because as somebody was saying in one of the previous meetings, writing a program to apply a theory is quite different from just talking about the theory. So then you say to yourself, well what's the program going to have to do, because of course it's got to do some language-using task? It's not enough to have a system which just takes some English input and prints out some lovely diagrams in colour and different sorts of typography that you can get nowadays with modern machines, and you look at that and say 'Yeah, that's the deep meaning of that sentence'. That won't do, and we know it won't do. So you look for a task which might actually evaluate it, and it may be the case, as Henry is suggesting, that translation is too hard a task, because we really don't have any idea about how to do anything. But the problem with translation is that in some ways it is a hard task, and therefore it's a good one, but we have this difficulty we mentioned earlier which is how to evaluate whether the translation's actually effective or not. So we have a problem there if our initial motivation is to evaluate a theory by building a computational program. The thing we want the program to do is itself rather hard to evaluate.

WHITELOCK: It's much easier surely to evaluate the translation than most other natural language tasks that yon can think of.

SPARCK JONES: Well is it any harder or more difficult Hum paraphrasing in the same language, for example?

THOMPSON: Yes – because people regularly do it and there are de facto standards, even if there are no analytic standards.

SPARCK JONES: Yes but we paraphrase all the time too.

THOMPSON: Not as a task that is evaluated. I think that paraphrasing (begging Bran [Boguraev]'s pardon) is an over-touted phenomenon.

KAPLAN: It also has the added pitfall that you've got to get it slightly different.

MOORE: As someone who has done no work in this area at all, I would like to hear some of the details of why being able to interpret one language and generate another is different from being able to translate. I mean I really just don't know enough about it.

JOHNSON: 'Interpret' in the non-technical sense, in the non-professional sense, you mean?

MOORE: 'Understand' if you like.

Brian CHANDLER: You can translate things you don't understand.

WHITELOCK: I think that's a red herring. If you're talking about putting together an understanding and a generating capability, then you're talking about some sort of carrier of meaning between the two. And that is presumably a static level of representation. Otherwise if it's a dynamic level, then it's being changed in some way, and you no longer have just understanding and generation. So it's a static device. You have to start talking about what should the representation in that static device be of a language value from a number system, say 'singular', where the number system is singular-plural: is that different or the same as the representation of 'singular' in a singular-dual-plural system? Or maybe there are five values for the interlingual values of features: 'singular' in a two-value system, 'singular in a three-value system, 'dual' and then 'plural' in the two systems. So what possible values can your features have in that interlingual representation? Either you say it's singular-dual-plural, and if the source language has a two-value system you just never generate 'dual', and you throw away the information about the distinction between 'plural' in a two-value and a three-value system, or you keep it, and it actually means then that just by looking at the interlingual representation, you could actually reconstruct what the source language that you came from was. Surely then what you've got isn't a language-free representation, if it's got all that information.

MOORE: I'm not sure I'm following this.

JOHNSON: Can I gloss that? In your terms, if indeed you do understand the source language, you understand it with respect to your requirements in the target language. And that's why it's different from just absorbing some text and then quite independently taking the sense of that text and regurgitating it somewhere else without any relationship whatsoever between those two linguistic systems. This I think is a generalization of Pete's point. I hope so.

MOORE: This may be very naive: let me concentrate on this number business. If you want to represent the singular-plural distinction in English in a meaning representation, presumably the language-neutral way to do that is in some formal terms to say what the truth-conditional consequences of using singular and plural in English are. Very naively in English that is that 'singular' means 'one' and 'plural' means 'more than one'. Now if you say 'Some men did x' and you translate that into a language-neutral meaning representation, you're going to have some representation of "There exists a set of men whose cardinality is greater than or equal to 2 ..." or something like that "... who did (such-and-such)'. Now you come along and you want to express that in a language that makes the singular-dual plural distinction ...

JOHNSON: Now you claim it's ambiguous, I guess.

MOORE: I don't know what good translators do in situations like this.

THOMPSON: But isn't that the crux of the matter, because now you are having to appeal to the question of what would translators do as opposed to what understanders of the language would do?

WHITELOCK: What they do is exploit every ounce of contrastive knowledge they have to look around for some way of getting round the problem or finding the answer, resolving the indeterminacy, or whatever. But it's contrastive.

MOORE: I don't know what you mean by 'contrastive'.

THOMPSON: It depends on an examination of the different expressive resources of the two languages concerned.

MOORE: Let me suggest a stronger hypothesis: that at this point you can forget about the original language. You've just got abstract meaning you want to express. There's something you want to say about a set containing at least two elements and the language you want to say it in is such that it doesn't let you say that that easily.

THOMPSON: I'm sympathetic to your views because I think I would have said that our positions were the same at the beginning of this discussion. I think something that's come from the people who spend their time on this is that, if I can paraphrase what Rod said a little while ago, multilingual paraphrase is not the same as translation. What you're talking about, and what I would have said translation was as well about an hour ago, is multilingual paraphrase, i.e. you take it in, and you regurgitate it, and it happens that on the way in it came in in one language and oh the way out it went out in another language. What these people are saying, and I guess I find it plausible, is that what a translator does is something more than that.

MOORE: I'm actually prepared to believe that. It's just that I would like some more convincing examples. I can't see that it makes any difference that in this case the information came in a certain language. It seems that you are in exactly the same situation in expressing a particular piece of information, taking the notion of information very abstractly, and expressing it in a language in which to express that information requires a fairly convoluted means.

KAPLAN: The problem I think is that the meaning is always under-determined by its expression in any one language. This is just generally true. There are many many things that could be said that are true of the situation that we're talking about, that are not expressed by any particular sentence. But going into another language, some of those things need to be expressed, but you don't know what they were. So you have to have some way of figuring out for the stuff that wasn't specified how you should actually fill in that information.

THOMPSON: Another good classic case is Russian to English article insertion: there are no definite or indefinite articles in Russian. People who translate from Russian to English have an extraordinarily complicated set of heuristics which are relevant to that task, which don't appear to figure in the task of understanding Russian, but do appear to figure in the task of translating Russian into English. How that could be is not clear to me, I must confess.

JOHNSON: I'm going to stop while Henry's on my side!

References

Carbonell, J., Cullingford, R.E. & Gershman, A.V. 1978. Knowledge-based Machine Translation. Research report 146. Yale University Department of Computer Science.

Chandioux, J. 1976. METEO: un système operationnel pour la traduction automatique des bulletins météorologiques destinés au grand public. **Meta** 21, 127-133.

Chomsky, N. 1982. **Lectures on Government and Binding**. Dordrecht: Foris.

Lytinen, S.L. & Schank, R.C. 1982. Representation and Translation. Research report 234. Yale University Department of Computer Science.

Montague, R. 1973. The Proper Treatment of Quantification In Ordinary English. In Hintikka, J., Moravcsik, J. & Suppes, P., eds. **Approaches to Natural Language**. Dordrecht: D. Reidel, 221-242. Reprinted in Thomason. ed., 247-270.

Slocum, J. 1984. Machine Translation: Its History, Current Status, and Future Prospects. **COLING 84: 10th International Conference on Computational Linguistics / 22nd Annual Meeting of the Association for Computational Linguistics** (Stanford, CA), Proceedings, 546-561.

Thomason, R.H., ed. 1974. **Formal Philosophy: Selected Papers of Richard Montague.** New Haven: Yale University Press.

Whitelock, P.J., Wood, M.McG., Chandler, B.J., Holden, N. & Horsfall, H.J. 1986. Strategies for Interactive Machine Translation: The Experience and Implications of the UMIST Japanese Project. **COLING 86: 11th International Conference on Computational Linguistics** (Bonn, West Germany), Proceedings. 329~334.

Wilks, Y. 1973. An Artificial Intelligence Approach to Machine Translation. In Schank, R.C. & Colby, K. M., eds. **Computer Models of Thought and Language,** San Francisco: W.H. Freeman, 114-151.