[*MT evaluation: basis for future directions*. Proceedings of a workshop...2-3 November 1992, San Diego]

PANEL: APPLES, ORANGES, OR KIWIS? CRITERIA FOR THE COMPARISON OF MT SYSTEMS

Introduction: *Muriel Vasconcellos,* Moderator
Pan American Health Organization

There are many different MT systems out there now—and a lot of different *kinds* of systems. Just as an example, there are 13 different commercial products that translate from English to Spanish. If you want to use a product that goes from English to Spanish, you're going to need to be able to distinguish between the features of those 13 that make it appropriate for *you.*

All too often MT evaluation exercises attempt to compare systems that were designed to do very different jobs. When they each have a different function, there's a limit to how meaningful a comparison can be. At the very best, the comparison of unlike entities is risky because there is no way to control the variables. At worst, a comparison of unlikes can be seriously misleading and do injustice to all—the comparees, the comparer, and the ultimate end-user.

The best way to deal with this problem is to evaluate MT as a *process* rather than a *product.* This approach has two very important advantages. First, it enables us to set up typologies for comparing apples with apples, oranges with oranges, and kiwis with kiwis. Hence the title of our panel. An evaluation of this kind cuts across glass box and black box. You may look inside the system, or you may look at it from the outside without getting into its inner workings, but the main point is that you're looking at a functional process that has a purpose, and then you're going to see that not all systems have the *same* purpose.

Here are some useful questions to ask:

Is the system designed to:

- Assimilate information or disseminate it?
- Handle most kinds of text or a limited domain?
- Cope with simple or complex linguistic structures?
- Translate from a single source language to multiple targets or vice versa?
- Process a large volume a day (10,000 words) or a small volume (less than 2,000 words).

Is it intended to be operated by:

- A monolingual user, a level 1 or 2 analyst, or a level 3-5 translator?
- A single user, multiple users on a mainframe, or multiple users on a LAN?
- A pre-editor, a user responding interactively, or a posteditor?

With a functional approach it is also easier to make predictions about a system's *potential*—after all, Isn't the whole point in evaluating a system to know what it's going to do for you tomorrow? Yesterday's and today's performances are nothing but snapshots in the life of a system. If you're thinking of spending the rest of your life with one, you're going to want to know how much care and feeding is involved. In his keynote, Yorick Wilks spoke of the type of evaluation in which you're trying to test how improvable a system is along certain dimensions, such as new, untuned subject matter. This is certainly the most mature test of a system. I have always maintained that unless we know the potential of an MT system, there's not much point in spending time to evaluate the formal output.

So today, instead of looking at particular texts in a particular series of test suites, we're going to look at the bigger, functional picture—an approach that better enables us to: (1) compare systems designed for different purposes, and (2) judge the potential of a system to live up to the purpose for which it was designed.

Now let's turn to our panel.

Panelists

Veronica Lawson, consultant, London: I thought it would be interesting to go back to the first study I did for the European Commission, which was a feasibility study on the machine translation of patents. I came to machine translation at that time as a translator. In 1978, I was the first translator in the UK to get involved in MT.

From my previous experience with the evaluation of human translation, it was clear to me that one had to reduce the subjectivity of evaluation in some way. In looking at MT, I realized that you really have to specify the use for which the translation is intended. Without some sort of specification of use, or purpose, it's very difficult to define any standard of translation, whether human or machine. (By *specification,* I mean a statement of what's required; *standards* are the means by which that specification is fulfilled.) Specifying the use gives you a restricted definition of translation for the circumstances concerned.

With this in mind, my evaluators in this patent feasibility study that I undertook for the European Commission back in 1978 were asked to assume that they were evaluating for scientific and technical information only—from a list of six options, namely, literary, legal, publication, information, selective translation, and summary or abstract. I went for "information" on these patent texts because I didn't see that they would be suitable for legal purposes, which is the other reason for which you would want to translate patents. (In fact, I was to learn in due course that some of my evaluators judged that the texts *were* useful for legal purposes, which was interesting, and in fact the last thing I expected.)

I was evaluating for three criteria: intelligibility, accuracy (we heard yesterday that these two usually correlate), and usefulness. Now, interestingly, in my study, intelligibility and accuracy did *not* correlate very well, and that is because patents, as you may know, are very low on intelligibility in the original. You can have an accurate translation which is not very intelligible but it truly reflects the original.

My report says that my method for measuring accuracy "in no circumstances is applicable to human translation, since the user's expectations of human translation and machine translation appear to be totally different."

Under the criterion of usefulness, I asked the evaluators to assess the usefulness of the output not only for information purposes but also for other patent uses (a variety of legal uses). Then the person I was working for at the European Commission said, "Ask them if this output is suitable for postediting." That was certainly a question worth asking, but of course it didn't work because it's very difficult to answer that question without knowing exactly what you're postediting for. Again, you would have to specify the use. One of my evaluators came up with a formula for answering this unanswerable question. You can do it, he said "You can do it, using a formula

$$\frac{L}{2} \times 2$$

where = the length of string." Suitable for postediting? It depends.

On another point, I have three questions I tell people to remember to ask if they see a demonstration of machine translation. Is the demonstration of random text? Have you brought it in your briefcase and handed it over? How much text has the system processed? Often they're very unwilling to answer that because it turns out to be very little. Many systems, especially research systems, and even some of the new commercial systems, haven't had a great deal of cumulative experience, and one therefore can't really form any conclusions about them. And finally, Can the system cope with not-found words? It must be able to do that.

***Eduard Hovy**:* I'd like to tell you my experience from being one of the people who helped develop the DARPA machine translation criteria. I will call this "functional" evaluation because that's a nice name, and also it happens to fit in with what Yorick was saying, and a little bit with what Henry Thompson was saying yesterday, both of whose talks made a great deal of sense to me.

If you think back to yesterday's panel—to what John White was describing about the DARPA evaluation—I wonder how much you really took away from that. What was the point? What did you learn? What did this tell you about the systems? Would you buy one of them? Would you look at the exercise from a research perspective? What was the value of this evaluation?

I myself, in spite of being one of the committee, was not happy with the way it came out—not because we didn't do very well, but because I don't think it was a very informative exercise. It doesn't tell me a great deal.

I want to tell you why I think evaluation is such a confused topic. We are mixing so many things together that we are having a panel called "Apples, Oranges, and Kiwis." The point is, we haven't yet begun to slice the problem up into pieces that make sense, so that we can say, "Oh, you're talking about *this* kind of evaluation as opposed to *that* evaluation, therefore *these* criteria apply, therefore I must think *this* way. This is what I'm going to get out of it. It *doesn't* apply to systems in this category." We have to do this if we want to make sense of this whole endeavor. I think we have to understand that there are *millions* of evaluations you can do, all kinds of things you can measure, and *what* evaluation you put together depends on *what* you want to get out of the evaluation.

Now, let's say one thing I might want to learn from the evaluation is, "How well could I do in the translation marketplace?" I'm talking obviously about commercial systems: it's the marketplace criterion. Will my system beat SYSTRAN out there in the marketplace? Now, that's fine for those people in the marketplace, right? It is not fine if I happen to be a small research person somewhere in a university with three students and I'm interested in Government and Binding or something. I'm not in the same league, I'm not in the same game. It doesn't make sense for me to evaluate my stuff against their stuff, because they have to worry about broad coverage and all kinds of things that are not germane to me.

Another question: How much of the core knowledge have you captured? This is an issue that has been being pushed lately, and it is actually a very interesting one. Supposedly, if we accept that there is a core lot of knowledge at the heart of the machine translation problem, and if we can just unpack that somehow and see what that is, then we'll be able to build all these marvelous systems. Then the evaluation question is "Forget the marketplace. We don't care. We need to find out how much this core knowledge is supposed to cover." The problem, as I see it, is that we don't *know* if there is a "core knowledge" to be looked at. There are so many different approaches to machine translation that the notion of one single, homogeneous body of core knowledge would not seem to be useful. A system that uses a symbolic message, such as PANGLOSS, or most of the systems that we build, cannot be compared in this regard against the system that Peter Brown was building originally when he used only pure statistics.

Now here's another question: How is my system improving? Forget all these other things; I just care about my own system. This is for the believers. I believe that my system is going to go somewhere, it's going to get there. Therefore I must add so many words in my lexicon and the new grammatical coverage so that it will be better next year, and therefore I'm going to get there. Right? Of course, there's no guarantee that you're going to get there, and this is what the doubters always say. "Why must I believe you?"

These three questions—I have another one—how your system stacks up against someone else's, but that's a muddy question—are my version of what Henry Thompson was saying yesterday so nicely when he referred to marketplace evaluation, assessment evaluation, and diagnostic evaluation (of your own internal growth).

Now, the question is, if you want to do all these different evaluations, how can you develop an overall evaluation that measures you for the marketplace? It seems to me, from all the reading I've done, that

there are a few major kinds of dimensions and there are lots and lots of subdimensions of these. The top one, clearly, is cost. I was reading some history of chemistry recently, and it seemed to me that, in spite of all the little fights that Lavoisier had about his theories with various people, eventually the results that worked were those what went out in the world and people could use it as science and make money off it and it became a real product. I think *eventually* the major criterion is going to be cost. The user one day is not going to care whether your system uses statistics or a mixture or whatever—it can even use a human in the loop—the user doesn't care, as long as it's cheap.

If we don't go for this overall measure of cost, let's look at one level down. Let's look at the different dimensions. Generality measures the domain size and coverage. Everyone has to pin it down and say "I'm going for a wide domain," or "I'm going for a very particular domain" (compare SYSTRAN and MÉTÉO, for instance).

Another one is the degree of automation. Some people will say, "I'm going to put human aids in," whether in the beginning, the middle, or afterwards. Others will say, "I'm going to not put any human aid in and go fully automatic," with emphasis on translation speed, and so on.

The third measure is aptness, which is quality and fidelity together. Often people break these out, but I wanted to put this all in three dimensions, so I put these two together. And usually, like in the DARPA MT evaluation, we have a measure for aptness. You will remember that we had a test for quality and a test for fidelity, which was the comprehension test.

So what we were measuring, in some sense, was time (which correlated with the degree of automation and other things) together with quality and fidelity.

Now, given that these (and others) are the kinds of dimensions you can measure along, the question is, you're sitting in this multidimensional space, and you know that you can put together millions of kinds of evaluations, and you know another thing—you know there are evaluations that you can put together that are suitable for systems you can buy (commercial ones, marketplace ones), or evaluations you can put together that are suitable for trying to compare your system against someone else's system, or there are ones that are internal for you (where you can count things). When you do what I consider the interesting one—trying to compare your system versus somebody else's system—the question is, How sensible *is* such an evaluation? And I think, if you try to put together systems that are so radically far apart, the amount of knowledge they rest on in common is so small that you can't really say anything sensible.

That's why I'm frustrated with the DARPA evaluation. If you say that the goal is high aptness, high automation, and high coverage, and you move predominantly in the direction of any one of these, the problems you're facing are different. In the one case, you might say, "I've got a fully automatic system and my problem is increasing the coverage." In another case you're saying, "My quality is high, I'm increasing the automation (as in the case of PANGLOSS), and it's not appropriate for me to be evaluated according to the other guy's criterion, because his problems aren't my problems and my problems aren't his problems."

To the extent that you share underlying knowledge, it makes sense to compare different systems. It seems to me there are three broad classes of machine translation systems about which it makes sense to say, "Okay, within this class or that one you can evaluate things and compare them sensibly. But cutting across classes doesn't really make sense because they're addressing different problems." Any more than you would attempt to compare a Rolls Royce, a dune buggy, and a bulldozer.

The first class—let's call our system the Scribe—is dissemination. The idea is: I'm producing some documents and I want a high-quality translation to go out into the world. I don't care if it takes longer to do it, I don't care if there's a human involved, but it's got to be high quality. And so the measures I need to apply here are quality and fidelity and the degree of automation. Right? Obviously.

Another kind of application we'll call the News Scout—where you have the translation of abstracts and news briefs and the idea is that I don't care if the quality is bad and I don't care if it's not super-accurate,

I just want to get a sense of what's going on there so that I can say, "Yes, get me this one," and then I give it to my other translator (my Scribe translator) and he brings me back the right thing. The measures here are the coverage (how general it is), the speed, and the fidelity. These are different measures—they're applying to a different problem.

Here's another one—the case of the PANGLOSS system, the one we're building. We've a human to help produce very high quality. Then there's the CANDIDE system.

Then there's the E-Mail Assistant, where you don't really produce a translated document; it's more like a behavior. Imagine you're at the computer and you see this e-mail message in another language, and based on your knowledge of the language itself and of the sender and the context, you can piece together the sense of the message enough well enough that you can do the appropriate thing. There you're looking at measures of speed and fidelity and you don't care about the other things. It's just got to be fast—fast and somewhat accurate.

These three systems compare to a bulldozer and Rolls Royce and a dune buggy and a motorcycle. They all get you from A to B, but they do it in different ways, along different dimensions of space, and so it's appropriate to measure them differently.

My argument, then, is that if you're designing an evaluation, it's important to understand what you're trying to do. You have to understand what it is you want to measure, how the space of evaluations breaks up. Choose your space, then choose your criteria appropriately, and then choose the systems within that set of criteria. Don't try to compare apples and oranges and kiwis, because it doesn't make sense.

Makoto Nagao, University of Kyoto: Evaluation factors are different for users and developers. For users there is a kind of black box evaluation, and the most important thing is that evaluation should be very easy. If the evaluation is very expensive, then probably, you can't do it. This is quite an important matter for users. Users' criteria for the purchase of MT use must be clear—if they are not clear, then they can't select any system.

The user data chart method was produced by Professor Nomura yesterday. To select the best system for the user's purposes, the user prepares a chart of his own requirements, which he than matches up against charts for various systems. Each of the prospective systems has its own chart, so by matching the charts against the user's requirements you can find the best-matched system. This approach is very good for users, but you must be very accurate in drawing the data charts. You have to set up various axes reflecting the user's demands. This can be done from the economic point of view, and it can also be done from the technological point of view. We believe that the data chart system is quite easy to understand and not very so difficult to interpret.

The next step is to have a trial and use of the system for several months, adjusting the mode of usage to obtain maximum efficiency.

For developers, I believe that the same type of approach can be used with a glass box test. The various facts must be as detailed as possible in order to test the ability of the prospective system. Linguistic factors, dictionary coverage, and, of course, the user interface, are all criteria to be taken into account.

This type of chart could be standardized for use by MT system manufacturers.

Ralph Dessau, Linguistic Products: My contribution here today is a little writeup about some considerations I feel have been overlooked in previous gatherings of this kind (see Annex A). Let me give you some of my thoughts behind this.

At Linguistic Products we cannot sell a product based on intelligibility. Our customers want *total* comprehension. On the other hand, we cannot provide that on the first pass—we don't even try. So our efforts have been to provide a system that is easy to *teach.* Early on, when we developed the first version of this translation (or "communication") software, we learned from large corporations that they could easily identify a relatively small vocabulary that frequently occurs in all their translations. For instance,

one major computer company identified 15,000 such phrases on top of about 5,000-10,000 customer-specific words. In order for them to machine-translate all the documentation that had to be translated, they had to have a system that they could teach 15,000 expressions, some grammar, some syntax perhaps, and then a vocabulary of perhaps 10,000 words. So the task, as far as we were concerned, was to create a system that would do this.

Recently these companies have come to us and asked, in addition, for their MT system to be able to handle formatting codes that will enable them to do tables, graphics, and other features required for desktop publishing. So that was another requirement that has nothing to do with linguistic ability or sensitivity, strictly a technical requirement. For patent translations, for instance, which is one of the applications that our software is used for, format is essential.

Moreover, it is not enough that you can translate or communicate, you must do it with the style of expression prescribed by the person who is paying for this work.

These are the things we have worked to incorporate in our system because we felt they were essential—much more than intelligibility on the first pass.

At the same time, technical advances have brought us the possibility of making these systems portable. We have created a system that fits on a notebook. You can take it anywhere. We have clients who travel all over the world with our system in their little notebook, and they can translate wherever they go. They want a system that is movable.

And another thing: it is no longer a question for most of our clients to translate into only one language. If it's an American company, they need to translate documentation simultaneously into seven or eight languages. If you have to work with one system for one language and another system for another language, it becomes a headache. So you've got to provide a whole spectrum—a base of languages that will cover the majority of the needs of these corporations.

Also, the range of uses for machine translation are broadening. For example, educational institutions are beginning to realize the need for machine translation. Not only at the university level, but also at the grade school level. Our schools today have a problem communicating with all the subcultures of different ethnic origin, and they must maintain communication with parents about what is going on in the school.

All these requirement should be taken into account in assessing a machine translation program and in formulating evaluation criteria.

Mike Tacelosky, MicroTac Software:[1] We publish the Language Assistant series—Spanish Assistant, French Assistant, German and Italian Assistant. We have over 100,000 users, so we are without doubt the least expensive and the most widespread PC-based machine translation software on the market.

My perspective of evaluating machine translation software is that of someone who is selling to a very large, broad-based market. The people who use our software, in most cases, are not using it to replace translation—they're using it to replace *non*translation. So our perspective for evaluating machine translation looks at whether or not it's capable of doing this job.

The software that we've had on the market for the last year (version 4) sells for $79 (at a Price Club and CompUSA it's closer to $40). It's a very inexpensive product. So the cost, compared with the other systems here, is basically free. So the question is, Will the product perform for its market? Does it meet the market's requirements? Based on the rate at which it has been selling, we would say right now that the answers are "Yes." We're not funded by outside capital right now, so in order to make payroll we *have* to sell to the market.

We've recently released version 5 of Spanish Assistant. We'll be coming out with the others (French, German and Italian) that will translate from the foreign language back into English. I suspect that that will provide a much more valuable resource to our customers, who by and large are replacing *non-*

[1] MicroTac Software will merge with Globalink, Inc., in October 1994.

translation. They have a document in Spanish, French, German, or Italian, and they need to understand it. The most important criterion is that the text be understand. If users are unsuccessful in understanding it, they can go on to a professional translator. If it's important that the translation be publication quality, as we talked about earlier in regard to patents, certainly a $79 system is *not* appropriate for patent translation—at all. A patent translation *absolutely* has to be a perfect translation—close isn't good enough. Because our system goes for the *non*translation, our marketplace is much more forgiving. If the end result is that you're able to understand something you weren't able to understand a few minutes ago, and the end users of our system by and large are not bilingual, that's what counts.

And I realize that the panel is running out of time and there are probably some questions, but I just wanted to just briefly explain what our software was and what the perspective of the PC (or at least MicroTac Software's) internal evaluation criteria is. And I realize that we'e not evaluated—out of 100,000 users, probably very, very few of them are in this room because PC-based machine translation systems for the broad market typically aren't evaluated next to $100,000 systems. Which is appropriate.

Muriel Vasconcellos: Incidentally, Mike Tacelosky tells me that MicroTac is already doing some parsing and they can produce a parse tree—which brings us to parsing. LOGOS has been doing a lot of parsing for a long, long time, and Bud Scott is going to mention some of the criteria that might affect a user in determining whether a system parses adequately.

Bud Scott: This morning we have been taking pretty much a black box perspective on the assumption that black box measures are of more interest to the user than anything about the internals of a system. However, as Muriel suggested this morning, one of the most vital facts about a machine system is it's capacity for growth. Certainly no user wants to espouse a machine that's not going anyplace, and the capacity of a system to grow has to do with its internals. It has to do with other factors, too, like how much money is being spent on it, what the sociology of the organization is that's developing it, and so on, but certainly it has to do with what's packed inside the machine.

Maybe we haven't talked much about the internals because it's rather hard to get a handle on the situation. I'd like to suggest one possible way to look at what goes on inside a machine in terms of the parse. The internals of the machine are typically thought of in machine translation as a mapping function, going from a string in language X to a string in language Y. But before you can do that, you have to know what you have in the X string. That's the function of the parse—to get at that string and specify it. And the mapping function can be no better than the specification of that string. That's what I'd like to talk about.

Here is a sentence that says, "The nurses keep clean sheets and blankets in them." It's a nothing sentence, but it's a great problem for a machine, as you'll see. I'll try to show you that there are a series of linguistic layers, or levels, that the machine has to deal with.

The first layer is the *lexical syntactic level.* Here the machine confronts a categorial ambiguity with respect to part of speech in the word "had".

In order to resolve that, we have to go to the next layer, which we can call the *sentential syntactic level.* In order to resolve the part-of-speech category, we have to see the category in the context of the sentence. This is the classic "bracketing" function of the parser. In doing this task, however, the parser discovers that new ambiguities are created at that level. For example, there is an adjective followed by two conjoined nouns, and the parser has to know, Does the adjective modify the first noun or both nouns?

To answer this question, we have to go down to the next layer, the *lexical semantic level.* We have to make use of semantic information that was available at the lexical level when the words were looked up. In this case, we can assume (statistically) that the adjective modifies both nouns because they are semantically homogeneous. But at the lexical semantic level there's yet another ambiguity that's introduced, namely the nuances, or the different meanings, of the verb "to keep."

And to resolve this problem we have to go to yet another level, which we can call the *sentential semantic level,* where the parser gets to look at the verb in its argument structure. Once it sees the argument structure, it realizes that the verb "to keep" is being used in the sense of "to store" rather than "to retain."

At this level yet another ambiguity is introduced—in reference to the pronoun. To resolve that, typically you have to go to the *extrasentential level.*

So you can see that there are a number of layers of linguistics that have to be dealt with by the parser, and you can classify systems on the basis of the extent to which they attempt to address these problems. I think a lot of the smaller systems certainly deal with the lexical syntactic and the sentential syntactic levels, but they are very light on the semantic aspects. There's a possibility of developing metrics for these also. I've talked to people who say, "Well, we've resolved 95% of the homographs in our system," and there are some special interest groups who are developing measures for bracketing sentences and measuring the success of bracketing. I think it would be possible to develop measures to test a systems ability to resolve ambiguities of words. You could develop an aptitude test for a machine translation system that would test it for its performance at each one of these levels. It would be very interesting for anybody thinking of buying a system to know its intelligence quotient (IQ) with respect to these kinds of problems.

Chris Miller*:* I recently conducted a survey of the systems that are commercially available in this country—whether through a distributor, directly in the stores, or directly from the developer. In fact, there aren't that many. I came up with a list of about 15 or 16 that run on PCs and/or workstations. I left in the workstation ones because most of them are pretty close to, or are already, working on a PC as well. I had a list of criteria that I started with, but it got changed *many* times. As you heard from each of the speakers, there are a lot of ways to look at them, and what I discovered was there's no easy way to say this system is better than that system. You really do have to look at what it is you're using it for. So the $79 system from MicroTac Software is something that's usable for many people out there—obviously, or they wouldn't have sold so many—and the systems that are more expensive are also being sold.

I wanted to expand on one thing that Ralph Dessau said when he talked about being able to import lists of vocabulary into a system. That's a feature that changes the viability of the system altogether. Some systems give better grammatical output on the first pass—for example, Globalink—but, on the other hand, if you have a system that will allow you to import an ASCII list of your customer-specific terms, you may get the job done in less time. So when you're purchasing a commercial system, you have to look at the cost and time involved, since the whole purpose of buying the system is to save time and money.

I didn't do very much linguistic evaluation in my study because I'm still waiting to hear what criteria are going to be fair.

General Discussion

• *(George Mallard,* Linguistic Products) This is a question for Chris Mill. How would you measure the adaptability of a system, and how important do you think that is?

• *(Miller)* What I did was to run through each of the systems from start to finish and see how long it took me to update the dictionaries in order to get the output I wanted. Of course, when it was easier to update the dictionary, the systems came out faster. So, from my perspective, the faster ones would be better, even though some of them weren't linguistically better than the ones that took longer. Which one would you rather own?

• *(Mallard)* Then what you're saying is that maybe the adaptability of the system may be more important than the actual core linguistics within the system?

• *(Miller)* Right. I'm an enthusiast of these products; I have them and play with them for fun. It's more fun for me to use one that has the features I want and is faster. I don't like having to spend hours to update the dictionary.

There's another thing I should say, too. *Every* system has to be customized. You don't get anything straight out of a box that you're going to find very useful. So how upgradable it is and how easy it is to use has to be weighed very carefully. If it takes you three months to learn as opposed to a few days, that's a big difference.

• *(Mallard)* So you're saying that it's important for these systems to be able to accept new terms and integrate them into their core linguistic system, whatever that may be.

• *(Miller)* Right. The other one Ralph mentioned—the desktop publishing—that's a big one, too, because if you have to reformat your output, that's hours and hours of work. If it will retain your WordPerfect formatting codes, your document is finished much faster.

• *(Mallard)* Exactly, so that would be reflected in your total cost. . .

• *(Vasconcellos)* I have heard that the McGraw-Hill publishing company considers that 50% of the translation cost is reformatting and reintroducing the format codes to SGML. If you can have that done for you, I think, a lot of users, in that situation, would buy an MT system just to get the codes in there and then worry about moving the text around, which would still save them money.

• *(Sergei Nirenburg,* Carnegie Mellon University) I have a question for Ed Hovy. And it's connected, in fact, to the previous question. Ed, you listed several types of measures, and the first one at the top was cost. But then you decided, very brusquely, that "no, we don't touch that." Well, why?

• *(Hovy)* I was hoping nobody would ask that question. I think cost is appropriate when your system reaches a point when you can think about selling it—that's the ultimate black box measure. But, for our DARPA purposes, we're interested more in the knowledge involved, so we want more of a glass box measure, and so, then, we have to go one step and we have to look at other dimensions. That is why I said cost, for our purposes, was not so useful.

• *(Nirenburg)* But that seems to be the only way to compare the systems across the ...

• *(Hovy)* Yes, that's my belief. That if you really want to compare such different systems, you have to go black box.

• *(Dessau)* I'd like to add to that comment. Our experience has been that the cost of the system—cost of the hardware, cost of the software—is negligible. What is expensive is the cost of the labor that you have to input into making the system work for you.

• *(Klaus Schopmeier,* Translingua) I would like to mention something that should enter into the equation, although it has *nothing* to do with machine translation whatsoever. I has to do with what software you use for postediting. If you have software that doesn't support such things as drag-and-drop editing, you're going to have a lot of trouble doing your postediting.

Another feature that comes in is the ability of translators to handle word processing. I find that sorely lacking in general. Also the willingness to learn it, because these people feel that they are in the realm of knowledge and therefore don't have to go down into the valley frequently and look up a lot of details in the manuals. Also, they're under production pressure, and the learning of the tool (which is essential for any craft) usually remains on the back burner.

• *(Vasconcellos)* And some products have a better learning curve—you can learn some products faster than others.

• *(Schopmeier)* Yes, it's true, but it seems basically that the mouse-driven word processors have a faster word-editing potential because of the directness of eye-hand (eye-action) coordination.

• *(Miller)* Marjorie Leon has macros she's put into WordPerfect. And one of the recommendations in my study is that developers also package macros to go with their system, so that the postediting is made easier.

• *(Chris Montgomery,* Language Systems, Inc.) On the subject of parsers, we did some work with Calspan Corporation, which is going to be a Rome Laboratory Air Force Technical Report. What we did was design a natural language evaluation procedure which could be used by anybody for their system. It isn't complete. However, it has the advantage that it's built to be system-independent and domain-independent. If anyone is interested in obtaining the report, I imagine it will be available through NTIS. You could also ask Sharon Walter at Rome Laboratories. She was the project manager, and I'm sure she would be happy to arrange to have you receive a copy of the report.

• *(Elliott Macklovitch,* CWARC) I'd like to take issue with a comment that Chris Miller made regarding the Language Assistant series. She said that the product must meeting some clients' needs, or MicroTac wouldn't have sold so many copies. I'm not sure that follows at all. I think what that shows us is that there's a desperate need for language-assisted software; whether the product corresponds to users' needs is another question. It's like selling snake oil solutions for baldness; just because they've sold so many cases of it doesn't mean they've solved the problem. The only way to know whether the $79 product satisfies users' needs is to check whether they're growing hair several months after they used it.

• *(Tacelosky)* I've recognized that within the translation industry the fact that our product is selling for $79 and is being presented as a machine translation system causes a lot of confusion—and, in some cases, hostility—on the part of people who are working on much more expensive systems.

The point made about evaluating whether or not the users are using the product is certainly valid. We're confident that our product is being used because of the way we survey our user base, the way we offer upgrades, and the way our users continue to upgrade and contact us about the product. I'm comfortable in responding to that.

I'm also comfortable with *not* specifically targeting the professional translating industry, because the product is *most* appropriate for *non*translators, as opposed to professional translators, which is the point I should have tried to make more emphatically when I spoke for the first time. Systems that are designed for professional translators to postedit have an entirely different set of criteria than systems such as ours that are designed for nonprofessional translators. The professional translators need both the accuracy and the postediting tools, and they may also need to have access to pre-editing techniques.

• *(Dessau)* May I ask you very directly: Do you monitor the users of your system? Do you have any way of knowing, several months after they purchase it, whether they continue to use it? Do you have any statistics on that?

How can we tell whether or not users like the product? We also offer a money-back guarantee and we have less than a 3% return rate.

• *(Theodora Landgren,* Bureau of Translation Services) Chris, how sure are you that you've located all the systems that are available? Where did you get your list? Did you include Japanese and Chinese systems in your survey?

• *(Miller)* How sure am I? Well, even the last day before I came here another system had just hit the market and called me, so I'm not completely sure, but I started with a list of 165—I was given lists that the European Community had put together, among others. But yes, there may be others out there.

• *(Landgren)* Okay. I guess my question is, if you started with 165 and ended up with 15, what happened to the other 150?

• *(Miller)* There are another 16 that are sold in Japan, but most of them run on workstations that are not really easily purchasable here, so they're not really in use in this country. I was trying to find things that were commercially in use in the States because that was my mandate. Looking for the systems was the hardest part. I spent several months.

• *(Landgren)* Another thing you said that I wondered about was that you were looking at these systems from the perspective of a user (I like this feature as opposed to that feature, and so on). It just concerned me a little that your conclusions might be subjective rather than objective.

• *(Miller)* Actually, I'm a big fan of all of the products and I think all of them are good for certain things, so there's no one product that I favor over any other to any great extent. I think they all have wonderful features, and the ideal system would be a combination of a lot of the features all lumped together in one. There's no single product that has all the features I like. And I should clarify that as a user I'm more of an MT enthusiast. I actually buy these things and have been following them for the last four years as a fan, even when there's no work to do on them, I still play with them.

• *(Landgren)* One last thing: you said "we're all looking to save time and money," and I'm not sure that that's correct . . .

• *(Miller)* For the commercial systems only. A commercial system's only purpose, really, is to save time and money. Otherwise the customer wouldn't be purchasing it.

• *(Vasconcellos)* I think that's a fair assumption. People wouldn't be using MT unless they were trying to save time or money. It may also improve uniformity if you're working on a large project—that's a possibility.

• *(Dessau)* I'd like to comment on that, if I may. Quality in the sense of accuracy is essential. That is what machine translation can contribute. No ambiguity, and a list of approved, established terminology that will inevitably come up, so that when you have several translators working on the same kinds of texts they will all stick by the same rules.

• *(Vasconcellos)* Ralph, let me tell you: I supervise 30 posteditors, and they change *anything they want to.* For me, the hardest part about using MT is helping translators to know what kinds of changes to make —when they should intervene, and when they should leave the output alone.

• *(Henry Thompson,* University of Edinburgh) On the comments by Professor Nagao and Chris Montgomery, I'd just like to enter a small caveat with respect to profiling of any kind, whether it's grammatical test suites or radar diagrams. Those both fall into the general category of what I called "diagnostic evaluation." Any kind of performance profile that says, "Here is some taxonomy of phenomena"—whether syntactic phenomena, aspects of user needs, or whatever else—"and here is a profile for a number of different systems. System A looks like this, and System B looks like this. . ." (Whether it's a radar diagram or you flatten it out doesn't change the information content.) The problem with those diagrams is that if you portray them as adequacy evaluation as opposed to diagnoses, I think you're being misleading.

Confronted with such a profile, you as a user would say, "Well, what does that mean? How do I know if passive is more important to me than embedded relatives?" The fact that System A does well over here and badly over there and that System B does well over there and badly over here—that doesn't help a user decide whether or not to buy something. So I think being able to produce pretty profiles in a diagnostic kind of evaluation may be very useful for developers—especially for iterative testing during development, to make sure you don't lose ground, as it were. But whether it's any use to people deciding whether or not this system if fit for a particular purpose, I beg to differ. I call into evidence a very interesting study done at the University of Essex of a couple of PC systems, where they got very neat performance profiles and were forced to admit that this made *no* contribution to choosing which system to buy for a particular purpose.

• *(Vasconcellos)* That's an interesting perspective. I thought that the point of the Japanese model was that you superimpose your needs over to various models and then you decide what's going to be right for you.

• *(Thompson)* Yes, but what's the significance of a mismatch? There are two mismatches on two different diagrams. How do you know which of those two mismatches actually contributes to your decision?

• *(Vasconcellos)* Aren't you the position of a subscriber to *Consumer Reports?* You let the buyer beware, you know where there is a match and where there isn't a match.

• *(Hovy)* Surely it depends on what the profile is. If it's just syntactic phenomena, sure, the user doesn't care, but if it's things like cost and speed and so on, then the user does care, in fact.

• *(Thompson)* Right. I think the point to be made is, absolutely, that the *Consumer Reports* model is the right model (I mentioned it yesterday—I like it), but it's important not to present these diagrams as if they determine a decision. They are a way of presenting information. But overlap (or lack of overlap) is not something you can then measure and use to come to a decision. You are left with a qualitative decision still to be made. There's a pseudo-quantitative aspect to the presentation of these diagrams, and I think that's what's misleading. Ed Hovy's absolutely right.

• *(Nagao)* If the user cannot decide, then he will not buy it.

• *(Montgomery)* I'd like to add just one quick comment to Henry Thompson's diatribe here. The objective of the benchmark test that I just talked about that we devised with Calspan is to produce just such a profile, and the assumption is that you *can* characterize the kinds of text you have (at least to some extent) in terms of what is useful. There's a *huge* difference between the kinds of parsing rules and

objective of the benchmark test that I just talked about that we devised with Calspan is to produce just such a profile, and the assumption is that you *can* characterize the kinds of text you have (at least to some extent) in terms of what is useful. There's a *huge* difference between the kinds of parsing rules and everything we use for the MUC kind of work that Beth Sundheim talked about yesterday versus the questions-and-answer kinds of things you get in an interview situation.

I think in most cases that you are capable of looking at whether what you're trying to design is some kind of an interface tool or some kind of a text-processing tool—and what are the kinds of sentences that occur in there—and take advantage of that. Of course, if you *can't* do that, then you can't use something that does your profiling for you. And ultimately, I think, as Professor Nagao said, then you're not going to buy it. But, if you are intelligent enough to see the kinds of text that you have and you have some variety in these and can characterize them in some way, then you can take advantage of such profiles.

• (Denis Gachot, SYSTRAN) I'd like to emphasize the importance of postediting. Two years ago our company decided to offer full postediting service. That means that our customers provide us with the document, we translate it with the machine, and we subcontract the postediting. And we certainly have learned a lot!

In the majority of cases we are asked to produce publishable documents. We have found that 60% of the cost goes into postediting as opposed to producing the machine translation, because producing the machine translation is the easy part. And I want to underscore what Klaus Schopmeier said earlier: if you don't have the right tools to do the postediting, this really can make or break a project using machine translation. We had to develop expertise in Framemaker, Interleaf, etc., etc. And more and more of our customers are asking advice because we now interface with nine various document-processing softwares out there. So once again, if you look at an MT system, don't underestimate the importance of postediting.

• *(Jackie Murgida,* FBIS) Everybody's been talking about human translation versus machine translation, and today we're talking about postediting as though human translators don't do any postediting of themselves, and as though other people don't postedit them. I've worked in situations where some translators need a lot of postediting. I'm not a full-time translator myself, but when I translate I postedit myself many times over—not just once or twice. So I would like to know, when we talk about 60% of the cost being in the postediting, Does anybody know how much time is spent in postediting and production for publication using human translation. With something like Arabic or Chinese, you pretty much take what you can get: you have mediocre to good translators, and you're lucky if you have some excellent ones.

• *(Vasconcellos)* My experience with the translation market is that it's self-selecting: the less money you offer, the poorer the translations you get.

• *(Margaret King,* ISSCO) I thought that somebody on the panel would say this, but I think it's important that somebody should say it: you have to distinguish between the amount of work that's going into revision or postediting—"thinking" kind of intellectual work and fixing the corrections—and the purely mechanical work of getting the formatting right and all that kind of thing. I think when the people on the panel have been talking about postediting, it's that kind of mechanical work that's been bothering them, not the intellectual work of trying to think out what the correction should be.