Yves Champollion was born 1956 in Paris, France; he is related to the famous Aegyptologist J-F Champollion. He has travelled extensively in the 70s and the 80s both to escape so-called higher studies and to quench his thirst for languages and cultures. Having learned enough languages and traveled enough miles, he met a Japanese girl in London, who turned into his fiancée in Germany, then turned into his wife in New York until both settled down in Paris. He got into the computing craze of the eighties, produced various pieces of software (such as "Le Tableur Comptable", an iconoclastic spreadsheet designed for accountants), then got into free-lance translation then into programming translation tools such as Wordfast—all for the sheer love and pleasure of it. Hobbies include traveling and learning languages (after French, Latin, German and English taught by the Fathers of Saint Vincent's College, to whom his heartfelt gratitude is hereby expressed, Spanish, Portuguese and Russian were added, as well as some Japanese and, last but not least, Shangana, a Zulu-related language of Southern Mozambique, where he sponsors a secondary school.

Mr. Champollion can be reached at yves@champollion.net

Machine translation (MT), and the future of the translation industry

by Yves Champollion

A usual concern among human translators is that machines will take over their business. On the one hand, prophets of doom announce a general crisis sending human translators onto the dole. On the other hand, any reasonable person who has tried MT software knows that human translation will be around for quite some time.

> The computer *computes* chess, while Kasparov *plays* chess. A computer will never *understand,* but it can *translate.*

For how long? Yes, the output from any MT software is still laughable. But beware. MT software is still in infancy. And, given the pace of development in the computer industry (both in software and hardware), we may see, sooner than expected, an MT solution that provides decent translation. All it takes is a resourceful computer and better MT software. The hardware is here. The software will inevitable follow. Still, the output may not be good enough for public display, so the question turns into: will the future of human translation be... *proofreading computer output*?

The bad news is yes. It all boils down to how long it will be until computers produce decent translations. All they need is basically here. Neural Networks and Artificial Intelligence are slowly becoming better. The discoveries in other (well-funded) industries that are large consumers of AI (civilian and military cybernetics, such as obstacle recognition, routing devices etc) will soon impact MT capabilities. Furthermore, the compilation of extensive knowledge bases such as dictionaries, glossaries, and translation memories will help improve machine translation.

As of going to press (November 2000), most of what MT software does is word-for-word translation followed by some grooming based on a set of rules. No surprise, the result is

From the Editor

TJ in the New Millennium
by Gabe Bokor

Index 1997-2001

Translator Profiles

A Vocation Found
by Danilo Nogueira

The Profession

The Bottom Line
by Fire Ant & Worker Bee

Existe-t-il des traducteurs spécialisés ?
by Frédéric Houbert

Translator Education

Revision—Food for Thought
by Nick Somers

Program Evaluation—A Missing Critical Link in Translator Training
by Moustafa Gabr

Spanish

Strategies for Translating "-ing" Adjectives and Nouns from English to Spanish
by Constanza Gerding Salas

---

barely readable.

Let's take a comparison with what humans do with, for instance, calculation. There are actually 3 ways (and maybe more) of performing a calculation:

1. **computation**. When asked the result for 145 + 133, we actually break down the operation into smaller ones, perform the necessary calculation and give the answer.
2. **memory**. When asked the result for 8 x 5, we immediately respond with recalling a table which we learned at school.
3. **common sense**. When asked whether 1,450,000 x 3,789 is greater or smaller than 1, we give the gut answer "greater," although we do not actually perform the calculation (a computer will not respond as we do—it will calculate first, then give a final answer).

Kasparov can confirm that computers use methods 1 and 2, with considerable speed. We may say that method 3 is nice and poetic, but not efficient. This is a serious mistake, however. All serious IT engineers (there are thousands of them) that are concerned with what computers will do next are working precisely on that third method.

In other words, they are working precisely on how to turn you, a translator, into a proofreader. This may take a long time, but don't rejoice too fast. A long time, in the world of IT, is 3 to 5 years.

So-called fuzzy logic can make some people laugh. Those who were around in the microcosm of computer freaks of the eighties remember that fuzzy logic and fuzzy processors were regular topics in discussion groups and specialized magazines, but they were conspicuously absent in the nineties, as if those dreams had failed to deliver anything solid.

Fuzzy logic in itself is not difficult to implement. Any serious programmer can program fuzzy logic, or even better, implement a neural network. Once the fuzzy processor, be it soft or hard, has delivered a set of options to a given problem, the problem rests entirely on choosing the most "reasonable" option. If this applies to chess, the answer is pretty straightfoward: reasonable means winning the game, period. In most human activities, however, and especially in language, the end purpose is not that simple. Consistency (human consistency) means that the answer has to match common sense, defined as the end result of countless learning situations which a person has lived since birth (we may distinguish one's personal trial-and-error situations, the wisdom acquired from education, plus the inborn instinctive knowledge: mature, nurture and nature). So the two-fold question is: can a computer memory store such a sum of knowledge? Can a program correctly process it and draw conclusions from it?

The answer to the first question, in the absolute sense, is no. The only way of *knowing* how ice cream tastes is to eat some. The only way of knowing how treason feels is to actually being

betrayed, and so on. A computer can store a description of such things, but it cannot harbor feelings.

The answer to the second question, in the absolute sense, is no. Interpreting a knowledge base to which the program is fundamentally alien will inevitably lead to nonsense.

But if we stop dreaming of man re-creating man through science, and take our expectations to a reasonable level—can a machine actually perform some human tasks with reasonable accuracy, the answer is an obvious yes, and the time it will happen is soon. A decent MT machine is just around the corner. The computer *computes* chess, while Kasparov *plays* chess. A computer will never *understand*, but it can *translate*, at least to some extent. And, since translation without understanding is meaningless, the future of the human translator is proof-sensing what a machine has pre-translated.