In “Languages and the Computing Profession” (The Profession, Mar. 2004, pp. 104, 102-103), Neville Holmes describes a method of automated language translation using a standardized “completely unnatural” intermediate language and discusses various problems. This method may work well for translating the literature of various technical fields because they have well-defined vocabularies.

The problems Holmes discusses are more serious in fields of human discourse outside the technical areas. Because human languages do not match well with regard to vocabulary, phrases, puns, and so forth, any translation that a human creates involves making subjective choices in translating words and other elements of the source language. These choices depend on the particular translator’s biases.

Even if computers perform the translations, a degree of subjectivity will be present in the translation software since it is unlikely that there could be a one-to-one mapping of the words, phrases, and so on in all human languages to the intermediate language.

In addition, for general literature, the characteristic of literality is problematic. Idioms, clichés, hackneyed phrases, and so forth do not have a counterpart in the destination language. These choices depend on the translator’s biases.

Holmes’s discussion of work to be done shows that he has thought about these matters. However, he does not explicitly discuss subjectivity. I am interested in knowing if he expects that subjectivity can be eliminated from the software.

To avoid, or at least lessen, the building of bias into the software was why I emphasized the importance of having philosophers (I had ethicists particularly in mind) and semanticians central to the project. Indeed, it is another reason for such a project to be under the aegis of the United Nations.

On the other hand, the bias that an author or reader inevitably imposes on text, even in technical fields, is wonderfully human, and the last thing I would want to do is eliminate it. That is why I suggested that departures from literality, perhaps the most obvious source of bias, might be encoded punctuitionally in the intermediate language.

Furthermore, my suggestion of adding “parameters that allow selection [and detection] of styles, periods, regionalities, and other variations” to translation programs would, for instance, provide for a document in English with one spectrum of biases to be translated into the intermediate language and then back into English with a completely different spectrum of biases.

What I am suggesting merges with interpretation in the long term, but there will be some texts that cannot be interpreted, only mimicked. One example is the kind of “Wockerjabby” doggerel that went the rounds quite a few years ago:

Eye halve a spell ling check err.  
Eat came whither peace see.

Eat plane lea marques form I revue.  
Mist ache sigh mite knot sea.  
I've run this pome threw eat,  
Aim shore yawp least two no.  
Its let err perfect inn it's weigh.  
My chequer tolld miso.

The techniques that Peter Maurer outlines in “Metamorphic Programming: Unconventional High Performance” (Mar. 2004, pp. 30-38) indeed have a successful history among software engineers emulating CPUs (or virtual machines) and creating fast state machines. The sources below provide additional explanations of the techniques as they are employed in various tasks:


As Maurer explains, there is performance to be gained by using procedural code. There may be two explanations for this. First, the label-as-value technique treats the compiler as a macro assembler, better matching how the underlying hardware works. Second, the performance ratios may be larger when using the GNU Compiler Collection.
Recent GCC versions have tended to produce slower code as their support for C++’s newer features has been expanded. Maurer’s benchmarks may reflect a temporary difference in effectiveness at compiling the two different types of constructs. In “Comparing C/C++ Compilers” (Dr. Dobbs J., Oct. 2003, pp. 12-24), Matthew Wilson provides a comparison of nine C++ compilers in terms of their performance, features, and tools.

Randall Maas
Chaska, Minn.
randym@ieee.org

MACHINE LANGUAGE TRANSLATIONS


I have seen machine translation horribly misused—for example, the manual of my OEM PC monitor has totally incomprehensible translations into several European languages. On the other hand, I know that some companies write user manuals in “controlled language” using English with a predefined limited vocabulary and limited grammar and syntax forms to which machine translation is applied with excellent results.

Andrew Bertallot
frengit@ieee.org

Correction

The biography for S.S. Iyengar (“Adaptive Neural Network Clustering of Web Users,” Apr. 2004, pp. 34-40) incorrectly listed him as a member the ACM and AAAS. He is a Fellow of these organizations as well as the IEEE.

We welcome your letters. Send them to computer@computer.org. Letters are subject to editing for style, clarity, and length.