Languages and the Computing Profession

Neville Holmes, University of Tasmania

Around Christmas, feeling the need for some light technical reading and having long been interested in languages, I turned to a story in *Computer*'s Technology News department (Steven J. Vaughn-Nichols, "Statistical Language Approach Translates into Success," Nov. 2003, pp. 14-16). Toward the end of the story, the following paragraph startled me:

> Nonetheless, the grammatical systems of some languages are difficult to analyze statistically. For example, Chinese uses pictographs, and thus is harder to analyze than languages with grammatical signifiers such as spaces between words.

First, the Chinese writing system uses relatively few pictographs, and those few are highly abstracted. The Chinese writing system uses *logographs*—conventional representations of words or morphemes. Characters of the most common kind have two parts, one suggesting the general area of meaning, the other pronunciation.

Second, most Chinese characters are words in themselves, so the space between two characters is a space between words. True, many words in modern Chinese need two and sometimes more characters, but these are compounds, much like English words such as *password, output,* and *software.*

Translating natural language is too important and complex for computing professionals to tackle alone.

Third, Chinese does have grammatical signifiers. Pointing a browser equipped to show Chinese characters at a URL such as www.ausdaily.net.au will immediately show a wealth of what are plainly punctuation marks.

Fourth, Chinese is an isolating language with invariant words. This should make it very easy to analyze statistically. English is full of prefixes and suffixes—the word *prefixes* itself has one of each—which leads to more difficult statistics.

I do not mean these observations to disparage the journalist who wrote the story—but they do suggest that some computing professionals may know less than they should about language.

LANGUAGE ANALYSIS

The news story contrasted two approaches to machine translation.

> Knowledge–based systems rely on programmers to enter various languages' vocabulary and syntax information into databases. The programmers then write lists of rules that describe the possible relationships between a language's parts of speech.
>
> Rather than using the knowledge-based system's direct word-by-word translation techniques, statistical approaches translate documents by statistically analyzing entire phrases and, over time, 'learning' how various languages work.

The superficial difference seems to be that one technique translates word by word, the other phrase by phrase. But what one language deems to be words another deems to be phrases—agglutinative languages mildly so and synthetic languages drastically so—compared to relatively uninflected languages like English. Also, the components of a phrase can be contiguous in one language and dispersed in another—as in the case of German versus English as Samuel Langhorne Clemens described (www.bdsnett.no/klaus/twain).

The underlying difference seems to be that the knowledge–based systems' data for each language comes from grammarians, while the statistical systems' data comes from a mechanical comparison of corresponding documents, the one a professional translation of the other.

LANGUAGE TRANSLATION

Looking at translation generally, the problem with the statistical approach is that it requires two translation programs for every pair of languages: one

Continued on page 102

Continued from page 104

going each way. Ab initio, the same is true of the grammatical approach.

The number of different languages is such that complete coverage requires numerous programs—101 languages would require 10,100 translation programs. Daunting when we consider the thousands of different languages still in popular use.

The knowledge–based or grammatical approach provides a way around this. If all translations use a single intermediate language, adding an extra language to the repertoire would require only two extra translation programs.

The news story does describe a similar approach, a *transfer system*, but this uses a lingua franca as the intermediate language, which in part is probably why it has been found unsatisfactory. The other unsatisfactory aspect is commercial—the extra stage when the commercial enterprise seeks merely to translate between two written languages adds extra complexity and execution time.

To cope with the variety of and within natural languages, a completely unnatural language must serve as the intermediary. Designing this intermediate language would be a huge and difficult task, but it would reap equally huge benefits.

Without this approach to machine translation, it would be difficult and expensive to cater for minor languages, to make incremental improvements as individual languages change or become better understood, and to add parameters that allow selection of styles, periods, regionalities, and other variations. When the translation adds conversion between speech and text at either end, adopting the intermediary approach will become more important, if not essential.

INTERMEDIATE LANGUAGE

The intermediate language must be like a semipermeable membrane that lets the meaning pass through freely while blocking idiosyncrasies. Although designing and managing the intermediary would be a nearly overwhelming task, certain necessary characteristics suggest a starting point.

- *Specificity.* Every primary meaning must have only one code, and every primary code must have only one meaning. The difficulty here is deciding which meanings are primary.
- *Precision.* A rich range of qualifying codes must derive secondary meanings from primary meanings and assign roles to meanings within their context.
- *Regularity.* The rules for combining and ordering codes, and for systematic codes such as those for colors, must be free from exceptions and variations.
- *Literality.* The intermediate language must exclude idioms, clichés, hackneyed phrases, puns, and the like, although punctuational codes could be used to mark their presence.
- *Neutrality.* Proper names, most technical terms, monocultural words, explicit words such as *inkjet* when used as shown here, and possibly many other classes of words must pass through the intermediate language without change other than, when needed, transliteration.

Global acceptance of an auxiliary language would foster the disappearance of minor languages.

My use of the term "code" in these suggested characteristics, rather than *morpheme* or *word*, is deliberate. Designing the intermediate language to be spoken as words and thus to serve as an auxiliary language would be a mistake.

First, designing the intermediate language for general auxiliary use would unnecessarily and possibly severely impair its function as an intermediary. Second, a global auxiliary language's desirable properties differ markedly from those needed for an intermediary in translation, as the auxiliary language Esperanto's failure in the intermediary role demonstrates.

Indeed, given the possibility of general machine translation, it is possible to make an argument against the very idea of a global auxiliary language. Natural languages—the essence of individual cultures—are disappearing much faster than they are appearing. Global acceptance of an auxiliary language would foster such disappearances. Versatile machine translation, particularly when speech-to-speech translation becomes practical, would lessen the threat to minor languages.

WORK TO BE DONE

Defining the intermediate language requires developing and verifying its vocabulary and grammar as suitable for mediating translation between all classes and kinds of natural language.

The *vocabulary*—the semantic structure, specifically the semes and their relationships—will in effect provide a universal semantic taxonomy. The semes would be of many different kinds, both abstract and concrete. A major challenge will be deciding which meanings are distinct and universal enough to warrant their own seme and where to place them in the seme hierarchy. The key professionals doing such work will be philosophers and semanticists.

The rules for associating and separating semes and seme clusters, the *grammar*, would encompass the work of punctuation, although much of the meaning found in natural-language punctuation could be coded in the intermediate language's semes, unless implied by the language's grammar. The intermediary grammar might need to designate some semes—for example, some of the two dozen or so meanings given for the term "the" in the *Oxford English Dictionary*—as required to be inferred if they are not present in the source language. The key professionals in this work will be translators, interpreters, and linguists.

When involved in a project to develop an intermediary language, these two groups of professionals will need to work closely together, as grammar and vocabulary are closely interdependent. In this case, both must cope with the translation of many hundreds of wildly different languages.

What role would computing professionals play in such a team? Given the project's purpose—to make general machine translation possible—computing professionals would be of vital importance, but in a supporting role. Using different approaches to evaluate the intermediate language and its use for a variety of languages would require a succession of translation programs.

Those involved in this project will need to consider how to keep Web pages in both their original language and the intermediary so that browsers could, if necessary, translate the page easily into any user's preferred language. Allied to this requirement would be consideration of how to index the intermediary text so that all of the Web's content would be available to searchers. Indeed, the qualities of an intermediate language could make search engines much more effective.

Translation of SMS messages and e-mail should also be studied; ultimately, use of the intermediate language in telephones for speech translation should become possible. Users would select the natural language to use on their phone. The translation might then be through text, staged with speech-to-text conversion, or the processor might convert speech directly to or from the intermediate language. In any case, intermediary codes would be transmitted between users' phones and thus the language of one user would be independent of another user's.

General use of such speech translation would trail text translation by a long way, but even general text translation would promote global cooperation, providing an excellent return on investment in the project.

I began this essay when reports from the UN Forum on the Digital Divide in Geneva first became public. The failure of this beanfeast was both predictable ("The Digital Divide, the UN, and the Computing Profession," *Computer*, Dec. 2003, pp. 144, 142-143), and a scandalous waste of money given the number of poor in the world dying daily of hunger or cheaply curable illnesses.

Strategically, a much better way to use digital technology to help the poor and counter global inequity and its symptomatic digital divide would be for the UN to take responsibility for the development and use of a global intermediate translation language. International support would be essential, both to make swift development possible and, more importantly, to protect the work from intellectual-property predators.

Success would make truly global use of the Internet possible. Ultimately, with translation and speech-to-text conversion built into telephones, UN and other aid workers could talk to the economically disadvantaged without human interpreters.

However, an intermediate language project such as this could not be contemplated without the strong and active support of various professional bodies, particularly those from the fields of computing, philosophy, and language. Computing professionals should work with others to get public attention for this project and ensure that the needed professional support is made available. ■

Neville Holmes is an honorary research associate at the University of Tasmania's School of Computing. Contact him at neville.holmes@utas.edu.au. Details of citations in this essay, and links to further material, are at www.comp.utas.edu.au/users/nholmes/prfsn.