# Why can't computers use English?

Written by Ray Jackendoff
Edited by Betty Birner

# If computers are so smart, why can't they use simple English?

Frankly, English isn't as simple as you might think. Although computers can do amazing things, what the human brain does in using English (or any other language) is even more complex.

For example, let's suppose we want to build a reading machine for the blind. The first problem we face, of course, is turning the written symbols on the page into speech sounds. This in itself is a difficult task, but let's suppose we can tell the computer how to pronounce every word in the dictionary. Even then, we will face many puzzles.

Consider, for instance, the four letters *read*; they can be pronounced as either *reed* or *red*. How does the machine know in each case which is the correct pronunciation? Suppose it comes across the following sentences:

**(1) The girls will read the paper. (_reed_)**

**(2) The girls have read the paper. (_red_)**

We might program the machine to pronounce *read* as *reed* if it comes right after *will*, and *red* if it comes right after *have*. But then sentences (3) through (5) would cause trouble.

**(3) Will the girls read the paper? (_reed_)**

**(4) Have any men of good will read the paper? (_red_)**

**(5) Have the executors of the will read the paper? (_red_)**

In sentence (3), *will* is not next to *read*, yet *read* is pronounced _reed_. In sentences (4) and (5), *will* **is** next to *read*, yet it's pronounced _red_. How can we program the machine to make this come out right?

First of all, the machine needs to know that in (1) and (2), *will* and *have* are 'auxiliary' verbs. This means that they modify the main verb *read* — in this case, by indi-

cating whether the reading occurs in the past or the future. So we could tell the computer that *read* is pronounced _reed_ after auxiliary *will*, and that it's pronounced _red_ after auxiliary *have*.

In (3), *will* is again an auxiliary verb modifying *read*, even though the two words aren't next to each other. So it should be pronounced _reed_. But in (4) and (5), *will* isn't an auxiliary verb at all; it's a noun. In these sentences, the auxiliary verb that modifies *read* is *have*, just as in (2), so *read* should be pronounced _red_.

How can the machine figure this out? It can't tell from the words alone, since *will* looks the same in sentences (3) through (5). Instead, it has to perform some sort of grammatical analysis of the text to find out which words fit together into phrases, and what category each word belongs to (noun, verb, etc.). To see just how complicated this can get, let's consider a few more cases:

**(6) Have the girls who will be on vacation next week read the paper yet? (_red_)**

**(7) Please have the girls read the paper. (_reed_)**

**(8) Have the girls read the paper? (_red_)**

Sentence (6) contains both *have* and *will* before *read*, and both of them are auxiliary verbs. But *will* modifies *be*, and *have* modifies *read*. In order to match up the verbs with their auxiliaries, the machine needs to know that *the girls who will be on vacation next week* is a separate phrase inside the sentence.

In sentence (7), *have* is not an auxiliary verb at all, but a main verb that means something like 'cause' or 'bring about'. To get the pronunciation right, the machine would have to be able to recognize the difference between a command like (7) and the very similar question in (8), which requires the pronunciation _red_.

We have very quickly gotten into detailed matters of grammatical analysis, just by trying to figure out how a machine could tell when to pronounce *read* as _reed_ and when to pronounce it as _red_ — something that any grade-school child knows. This particular issue may not seem important, but problems like this one

come up all the time in any computer application that uses human language, from reading machines for the blind to the automated phone system that tells you when your library books are due.

The _reed_ vs. _red_ problem is a very simple example of the countless puzzles that require the tools of syntactic analysis developed by linguists. Over the last three or four decades, advances in syntactic theory have given us a much better understanding of grammatical constructions — in English and many other languages — than we ever had before. These breakthroughs have made it possible for the first time for computers to use 'natural' human language, at least in some limited ways — for example, to translate documents from one language to another. But even the simplest use of language requires a vast amount of linguistic knowledge to be programmed into the computer, as the _reed_/_red_ problem shows.

As difficult as these questions are for computers, humans solve many such problems every time we read, write, talk, or listen. And we do it effortlessly, without even noticing the complexity of what we're doing, and certainly without knowing consciously how we do it.

Recent advances in psychology and neuroscience have done a great deal to improve our understanding of how the brain performs these tasks. Research into the inner workings of language structure has already given us perhaps the most detailed and precise analysis yet known for any task carried out by the brain, but there is still much more to learn. What is clear is that when it comes to human language, even the most advanced computer is currently no match for the abilities of the human brain.