Mother Tongue To Boolean

ince its early 1950s beginnings, computer-based natural language understanding has been inextricably linked to theoretical linguistics, a young science no older than computing itself.

And one of the crucial lessons so far learnt from both disciplines is the measure of just how complex human language is. Human beings understand each other with such ease that it's hard to see the machine's problem - until you try to write a computer program to imitate that understanding.

Early research, based on a blithe underestimation of the complexity of language, was crude. The pioneers of computational linguistics assumed, for instance, that fully automatic machine translation would require little more than a bilingual dictionary and a few rules of inflection.

But steady progress has been made. The commercial products having made their appearance during the past decade all do their best to address the major input text stumbling blocks of ambiguity, vagueness, and ill-formedness.

The spur to succeed lies in the enormous potential benefits of computers that can understand natural language. If your computer can talk to you in your own native tongue, then such applications as database consultation and computer-aided teaching will be made both easier and more times more effective.

And the understanding of full written texts will be the basis for such applications as machine translation, expert systems, and online knowledge bases.

Understanding is itself a nebulous concept. What does it mean for a computer to "understand?" In a paper published in 1950, not long before his death, computing pioneer Alan Turing suggested that behavior was what counted. If a computer responded to a piece of linguistic input just as a human would do in the same situation, then it had understood it.

While no one would suggest that today's natural language understanding programs are as intuitive as humans - certainly, they lack the breadth and flexibility of human understanding - they can nevertheless perform their tasks well, even if limited to narrow domains.

FIRST PARSE

A basic natural language understanding system typically contains a parser and grammar, a semantic interpreter, and an application module. In addition, it needs a lexicon or dictionary, and possibly a knowledge base.

The job of the parser is to analyze the input's syntax, creating a tree structure that shows the part of speech of every word in it and how each word fits into the larger structure of the sentence - much as used to be taught in secondary school English. For example, figure 1 shows the result of parsing the sentence "Peter Piper picked a peck of pickled peppers."

The system's parser should be independent of any one language. It is given a grammar and lexicon for each language upon which it will operate. The way grammar is represented in the computer varies according to the parsing method, but most methods explicitly or implicitly use rules of constituency.

For example, a very simple grammar might say that in English, a sentence (S) is made up of a noun phrase (NP), followed by a verb phrase (VP), and a verb phrase is a verb (V) followed by a noun phrase. This might be written:

\[ S \rightarrow NP + VP \]

\[ VP \rightarrow V + NP \]

Given, in turn, rules defining what constitutes a noun phrase, and a lexicon that will tell you that "pick" is a verb, a parser can use these rules to analyze the sentence in figure 1.

Parsing has received considerable attention in the past twenty years, and has developed in sophistication along with
It's one thing to be able to use your computer to search for a word in a termbank or typeset a text. It's quite another for your computer to determine the meaning of your text.

Research into computer understanding of human, or natural, language has been going on for the past thirty years - though it's only during the past decade that products making use of NLU have appeared on the market.

How far are we in getting computers to understand the flawed inconsistencies of human language and translate them into the Boolean formulas of their own "native tongue"? To cast a long eye on the current state of natural language understanding, Electric Word turned to Graeme Hirst, head of NLU research at the University of Toronto.

Typically, such interfaces parse the input, translate it into a database query language, and send it to the database, whose response is then presented to the user. The kinds of questions it will...