DICTIONARIES OF THE MIND

George A. Miller
Department of Psychology
Princeton University
Princeton, NJ 08544, USA

ABSTRACT

How lexical information should be formulated, and how it is organized in computer memory for rapid retrieval, are central questions for computational linguists who want to create systems for language understanding. How lexical knowledge is acquired, and how it is organized in human memory for rapid retrieval during language use, are also central questions for cognitive psychologists. Some examples of psycholinguistic research on the lexical component of language are reviewed with special attention to their implications for the computational problem.

INTRODUCTION

I would like to describe some recent psychological research on the nature and organization of lexical knowledge, yet to introduce it that way, as research on the nature and organization of lexical knowledge, usually leaves the impression that it is abstract and not very practical. But that impression is precisely wrong; the work is very practical and not at all abstract. So I shall take a different tack.

Computer scientists -- those in artificial intelligence especially -- sometimes introduce their work by emphasizing its potential contribution to an understanding of the human mind. I propose to adopt that strategy in reverse: to introduce work in psychology by emphasizing its potential contribution to the development of information processing and communication systems. We may both be wrong, of course, but at least this strategy indicates a spirit of cooperation.

Let me sketch a general picture of the future. You may not share my expectations, but once you see where I think events are leading, you will understand why I believe that research on the nature and organization of lexical knowledge is worth doing. You may disagree, but at least you will understand.

Some Technological Assumptions

I assume that computers are going to be directly linked by communication networks. Even now, in local area networks, a workstation can access information on any disk connected anywhere in the net. Soon such networks will not be locally restricted. The model that is emerging is of a very large computer whose parts are geographically distributed; large corporations, government agencies, university consortia, groups of scientists, and others who can afford it will be working together in shared information environments. For example, someday the Association for Computational Linguistics will maintain and update an exhaustive knowledge base immediately accessible to all computational linguists.

Our present conception of computers as distinct objects will not fade away -- the local workstation seems destined to grow smaller and more powerful every year -- but developments in networking will allow users to think of their own workstations not merely as computers, but as windows into a vast information space that they can use however they desire.

Most of the parts needed for such a system already exist, and fiber optic technology will soon transmit broadband signals over long distances at affordable costs. Putting the parts together into large, non-local networks is no trivial task, but it will happen.

Computer scientists probably have their own versions of this story, but no special expertise is required to see that rapid progress lies ahead. Moreover, this development will have implications for cognitive psychology. However the technological implementation works out, at least one aspect raises questions of considerable psychological interest: in particular, how will people use it? What kind of man-machine interface will there be?

What might lie "beyond the keyboard," as one futurist has put it (Bolt, 1984), has been a subject for much creative speculation, since the possibilities are numerous and diverse. Although no single interface will be optimal for every use, many users will surely want to interact with the system in something reasonably close to a natural language. Indeed, if the development of information networks is to be financed by those who use them, the interface will have to be as natural as possible -- which means that natural language processing will be a part of the interface.

Natural Language Interfaces

Natural language interfaces to large knowledge bases are going to become generally available. The only question is when. How long will it take? Systems already exist that converse and answer questions on restricted topics. How much remains to be done?

Before these systems will be generally useful, three difficult requirements will have to be met. An interface must: (1) have access to a large, general-purpose knowledge base; (2) be able to deal with an enormous vocabulary; (3) be able to reason in ways that human users find familiar. Other features would be highly desirable (e.g., automatic speech recognition, digital processing of images, spatially distributed displays of information), but the three listed above seem critical.

Requirement (1) will be met by the creation of the network. How a user's special interests will shape the organization of his knowledge base and his locally resident programs poses fascinating problems, but I do not understand them well enough to comment. I simply assume that eventually every user can have at his disposal, either locally or remotely, whatever data bases and expert systems he desires.

Requirement (3), the ability to draw inferences as people do, is probably the most difficult. It is not likely to be "solved" by any single insight, but a robust system for revising belief structures will be an essential component of any satisfactory interface. I believe that psychologists and other cognitive scientists have much to contribute to the solution of this problem, but the most promising work to date has been done by computer scientists. Since I have little to say about the problem other than how difficult it is, I will turn instead to requirement (2), which seems more tractable.

THE VOCABULARY PROBLEM

Giving a system a large vocabulary poses no difficulty in principle. And everyone who has tried to develop systems to process natural language recognizes the importance of a large vocabulary. Thus, the vocabulary problem looks like a good place to start. The dimensions of the problem are larger than might be expected, however, so there has been some disagreement about the best strategy.

If, in addition to understanding a user's queries, the system is expected to understand all the words in the vast knowledge base to which it will have access, then it should probably have on the order of 250,000 lexical entries: at 1,000 bytes/entry (a modest estimate), that is 250 megabytes. Since standard dictionaries do not contain many of the words that are printed in newspapers (Walker & Amsler, 1984), another 250,000 megabytes would probably be required for proper nouns. Since I am imagining the future, however, I will assume that such large memories will be available inexpensively at every user's workstation. It is not memory size per se that poses the problem.

The problem is how to get all that information into a computer. Even if you knew how the information should be represented, a good lexical entry would take a long time to write. Writing 250,000 of them is a daunting task.

No doubt there are many exciting projects that I don't happen to know about, but on the basis of my perusal of the easily accessible literature there seem to be two approaches to the vocabulary problem. One uses a machine-readable version of some traditional dictionary and tries to adapt it to the needs of a language processing system. Call this the "book" approach. The other writes lexical entries for some fragment of the English lexicon, but formulates those entries in a notation that is convenient for computational manipulation. Call this the "demo" approach.

The book approach has the advantage of including a large number of words, but the information with each word is difficult to use. The demo approach has the advantage that the information about each word is easy to use, but there are usually not many words. The real problem, therefore, is how to combine these two approaches: how to attain the coverage of a traditional dictionary in a computationally convenient form.

The Book Approach

If you adopt the book approach, what you want to do is translate traditional dictionary entries into a notation that makes evident to the machine the morphological, syntactic, semantic, and pragmatic properties that are needed in order to construct interpretations for sentences. Since there are many entries to be translated, the natural solution is to write a program that will do it automatically. But that is not an easy task.

One reason the translations are difficult is that synonyms are hard to find in a conventional dictionary. Alphabetical ordering is the only way that a lexicographer who works by hand can keep track of his data, but an alphabetical order puts together words with similar spellings and scatters haphazardly words with similar meanings. Consequently, similar senses of different words may be written very differently; they may be written at different times and even by different people. (For example, compare the entries for the modal verbs 'can,' 'must,' and 'will' in the Oxford English Dictionary.) Only a very smart program could appreciate which definitions should be paraphrases of one another.

Another reason that the translations are difficult is that lexicographers are fond of polysemy. It is a mark of careful scholarship that all the senses of a word should be distinguished; the more careful the scholarship, the greater the number of distinctions.

When dictionary entries are taken literally the results for sentence interpretation are ridiculous. Consider an example. Suppose the language processor is asked to provide an interpretation for some simple sentence, say:

"The boy loves his mother."

And imagine it has available the text of Merriam-Webster's Ninth New Collegiate Dictionary. Ignoring sub-senses:

"the" has 4 senses,
"boy" has 3,
"love" has 9 as a noun and 4 as a verb,
"his" has 2 entries, and
"mother" has 4 as a noun, 3 as an adjective, 2 as a verb.

Such numbers invite calculation. If we assume the system has a parser able to do no more than recognize that "love" is a verb and "mother" is a noun, then, on the basis of the literal information in this dictionary, there are 4x3x4x2x4 = 384 candidate interpretations. This calculation assumes minimal parsing and maximal reliance on the dictionary. Of course, no self-respecting parser would tolerate so many parallel interpretations of a sentence, but the illustration gives a feeling for how much work a good parser does. And all of it is done in order to "disambiguate" a sentence that nobody who knows English would consider to be the least ambiguous.

Synonymy and polysemy pose serious problems, even before we raise the question of how to translate conventional definitions into computationally useful notations. Any system will have to cope with synonymy and polysemy, of course, but the book approach to the vocabulary problem seems to raise them in acute forms, while providing little of the information required to resolve them. With sufficient patience this approach will surely lead to a satisfactory solution, but no one should think it will be easy.

The Vocabulary Matrix

As presented so far, synonymy and polysemy appear to be two distinct problems. From another point of view, they are merely two different ways of looking at the same problem.

In essence, a conventional dictionary is simply a mapping of senses onto words, and a mapping can be conveniently represented as a matrix: call it a vocabulary matrix. Imagine a huge matrix with all the words in a language across the top of the matrix, and all the different senses that those words can express down the the side. If a particular sense can be expressed by a word, then the cell in that row and column contains an entry; otherwise it contains nothing. The entry itself can provide syntactic information, or examples of usage, or even a picture -- whatever the lexicographer deems important enough to include. Table 1 shows a fragment of a vocabulary matrix.

Table 1. Fragment of a Vocabulary Matrix

Columns represent modal verbs; rows represent modal senses; 'E' in a cell means the word in that column can express the sense in that row.

	WORDS				
SENSES	can	may	must	should	will
be able to	E	.	.	.	.
be permitted to	E	E	.	.	.
be possible	E	E	.	.	.
be obliged to	.	.	E	.	.
certain to be	.	.	E	.	E
be necessary	.	.	E	E	.
expected to be	.	.	E	E	E

Several comments should be made about the vocabulary matrix.

First, it should be apparent that any conventional dictionary can be represented as a vocabulary matrix: simply add a column to the matrix for every word, and add a row to the matrix for every sense of every word that is given in the printed dictionary. (A lexical matrix can be viewed as an impractical way of printing a dictionary on a single, very large sheet of paper.)

Second, entering such a matrix consists of searching down some column or across some row. So a vocabulary matrix can be entered either with a word or with a sense. Thus, one difference between conventional dicticnaries, which can be entered only with a word, and the dictionary in our mind, which can be entered with either words or senses, disappears when dictionaries are represented in this more abstract form.

Third, if you enter the matrix with a sense and search along a row, you find all the words that express that sense. When different words express the same sense, we say they are synonymous. On the other hand, if you enter the matrix with a word and look down that column, you find all the different senses that that word can express. When one word can express two or more senses, we say that it is ambiguous, or polysemous. Thus, the two great complications of lexical knowledge, synonymy and polysemy, are seen as complementary aspects of a single abstract structure.

Finally, since the vocabulary matrix serves only to represent the mapping between the two domains, it is free to expand as new words, or new senses for familiar words, are added. Of course, the number of columns is relatively fixed by the size of the vocabulary, so the major degrees of freedom are in deciding what the senses are and how to represent them.

The Demo Approach

When the question is raised of what a computationally useful lexical entry should look like, it is time to shift from the book approach to the demo approach, where serious attempts have been made to establish a conceptual notation in which semantic interpretations can be expressed for computational use.

By "the demo approach" I mean the strategy of building a system to process language that is confined to some well defined content area. Since language processing is a large and difficult enterprise, it is sensible to begin by trying out one's ideas in a small way to see whether they work. If the ideas don't work in a limited domain, they certainly won't work in the unlimited domain of general discourse. The result of this approach has been a series of progressively more ambitious demonstration programs.

Among those who take this approach, two extremes can be distinguished. On the one hand are those who feel that syntactic analysis is essential and should be carried, if not to completion, then as far as possible before resorting to semantic information. On the other hand are those who prefer semantics-based processing and consider syntactic criteria only when they get in trouble.

The difference is largely one of emphasis, since neither extreme seems willing to rely totally on one or the other kind of information, and most workers would probably locate themselves somewhere in the middle. Since I am concerned here with the lexical aspects of language comprehension, however, I shall look primarily at semantics-based processing.

Vocabulary Size

Most of these demos have small vocabularies. It is surprising how much you can do with 1,500 well chosen words; a demo with more than 5,000 words would be evidence of manic energy on the part of its creator. A few thousand lexical entries have been all that was required in order to test the ideas that the designer was interested in.

The problem, of course, is that writing dictionary definitions is hard work, and writing them in LISP doesn't make it any easier. If you are satisfied with definitions that take five lines of code, then, obviously, you can build a much larger dictionary than if you try to cram into an entry all the different senses that are found in conventional dictionaries. But even with short definitions, a great many have to be written.

If you want the language processor to have as large a vocabulary as the average user, you will have to give it at least 100,000 words. One way to get a feeling for how many words that is is to translate it into a rate of acquisition. Several years ago I looked at Mildred Templin's (1953) data that way. Templin measured the vocabulary size of children of average intelligence at 6, 7, and 8 years of age. In two years they acquired 28,300 - 13,000 = 15,300 words, which

averages out to about 21 words per day (Miller, 1977).

Most people, when they hear that result, confess that they had no idea that children are learning new words at such a rapid rate. But the arithmetic holds just as well for computers as for children. If you want the language processor to have a vocabulary of 100,000 words, and if you are willing to spend ten years putting definitions into it, then you will have to put in more than 27 new definitions every day.

How far from this goal are today's demos? The answer should be simple, but it's not. It is hard to tell exactly how many words these systems can handle. Definitions are usually written in terms of a relatively small set of semantic primitives, and the inheritance of properties is assumed wherever possible. The goal, of course, is to create an unambiguous semantic representation that can be used as input to an inferencing system, so the form of these representations is much more important than their variety, at least in the initial experiments. In the hands of a clever programmer, a few hundred semantic primitives can really do an enormous amount of work.

Although it is often assumed that the fewer semantic primitives a system requires, the better it is, in fact there seems to be little advantage to keeping the number small. When the number of primitives is small, definitions become long permutations of that small number of different atoms (Miller, 1978). When the set of primitives gets too small, definitions become like machine code: the computer loves them, but people find them hard to read or write.

Combining Book and Demo

How large a set of semantic primitives do we need? It is claimed that Basic English can express any idea with only 850 words, but that really cuts the vocabulary to the bone. The Longman Dictionary of Contemporary English, which is very popular with people learning English as a second language, uses a constrained vocabulary of about 2,000 words (plus some specialized terms) to write its definitions.

Using the LDOCE as a guide, Richard Cullingford and I tried to estimate how much effort would be involved in creating a computationally useful lexicon. Our initial thought was to write LISP programs for 2,000 basic terms, then use Cullingford's language processor (Cullingford, 1985) to translate all of the definitions into LISP. We quickly realized, however, that the 2,000 words are polysemous; different senses are used in different definitions. As a rough estimate, we thought 12,000 basic concepts might suffice.

An examination of the LDOCE definitions also indicated that a great deal of information might have to be added to the translated definitions. Many of the simpler conceptual dependencies (information required for disambiguation, as well as for drawing inferences; Schank, 1975) have to be included in the definitions. Each translated definition would have to be checked to see that all sense relations, predicate-argument structures, and selectional restrictions were explicit and correct, and a wide variety of pragmatic facts (e.g., that "anyhow" in initial position signals a change of topic) would probably have to be added.

We have not undertaken this task. Not only would writing 12,000 definitions (and checking out and supplementing 50,000 more) require a major commitment of time and energy, but we do not have Longman's permission to use their dictionary this way. I report it, not as a project currently under way, but simply as one way to think about the magnitude of the vocabulary problem.

So the situation is roughly this: In order to have natural language interfaces to the marvellous information sources that will soon be available, one thing we must do is beef up the vocabularies that natural language processors can handle. That will not be an easy thing to accomplish. Although there is no principled reason why natural language processors should not have vocabularies large enough to deal with a any domain of topics, we are presently far from having such vocabularies on line.

THE SEARCH PROBLEM

As we look ahead to having large vocabularies, we must begin to think more carefully about the search problem.

In general, the larger a data base is, the longer it takes to locate something in it. How a large vocabulary can be organized in human memory to permit retrieval of word meanings at conversational rates is a fascinating question, especially since retrieval from the subjective lexicon does not seem to get slower as a person's vocabulary gets larger. The technical issues involved in achieving such performance with silicon

memories raise questions I understand only well enough to recognize that there are many possibilities and no easy answers. Instead of speculating about the computer, therefore, I will take a moment to marvel at how well people manage their large vocabularies.

In the past fifteen years or so a number of cognitive psychologists have been sufficiently impressed by people's lexical skills to design experiments that they hoped would reveal how people do it. This is not the time to review all that research (see Simpson, 1984), but some of the questions that have been raised merit attention.

Psychologists have considered two kinds of theories of lexical access, known as search theories and threshold theories.

Search theories assume that a passive trace is stored in the mental lexicon and that lexical access consists of matching the stimulus to its memory representation. Preliminary analysis of the stimulus is said to generate a set of candidates, which is searched serially until a match is found.

Threshold theories claim that each sense of every word is an independent detector waiting for its features to occur. When the feature count for any sense gets above some threshold, that sense becomes conscious.

Both kinds of theories can account for most of the experimental data, but not all of it -- which is unfortunate, since a clear decision in favor of one or the other might help to resolve the question of whether lexical access involves a serial processor with search and retrieval, or a parallel processor with simple activation. Since the brain apparently uses slow and noisy components, something searching in parallel seems plausible, but such devices are not yet well understood.

Accessing Ambiguous Words

Some of the most interesting psychological research on lexical access concerns how people get at the meanings of polysemous words. These studies exploit a phenomenon called priming: when a word in a given lexical domain occurs, other words in that domain become more accessible.

For example, a person is asked to say, as quickly as possible, whether a sequence of letters spells an English word. If the word DOCTOR has just been presented, then NURSE will be recognized more rapidly than if the preceding word had been unrelated, like BUTTER (Meyer & Schvaneveldt, 1971; Becker, 1980). The recognition of DOCTOR is said to prime the recognition of NURSE.

This lexical decision task can be used to study polysemy if the priming word is ambiguous, and if it is followed by probe words appropriate to its different senses.

For example, the ambiguous prime PALM might be followed on some occasions by HAND and on other occasions by TREE. The question is whether all senses of a polysemous word are activated simultaneously, or whether context can facilitate one meaning and inhibit all others.

Three explanations of the results of these experiments are presently in competition.

Context dependent access--Only the sense that is appropriate to the context is retrieved or activated.

Ordered access--Search starts with the most frequent sense and continues serially until a sense is found that satisfies the context.

Exhaustive access--Everything is activated in parallel at the same time, then context selects the most appropriate sense.

At present, exhaustive access seems to be the favorite. According to that theory, disambiguation is a post-access process; the access process itself is a cognitive "module," automatic and insulated from contextual influence. My own suspicion is that none of these theories is exactly right, and that Simpson (1984) is probably closer to the truth when he suggests that multiple meanings are accessed, but that dominant meanings appear first and subordinate meanings come in more slowly and then disappear.

Psychological research on lexical access is continuing; the complete story is not yet ready to be told. One aspect of the work is so obvious, however, that its importance tends to be overlooked.

Semantic Fields

The priming phenomenon presupposes an organization of lexical knowledge into patterns of conceptually related words, patterns that some linguists have called semantic fields. Apparently a semantic field can fluctuate in accessibility as a whole.

I have generally taken the existence of semantic fields as evidence in favor of theories of semantic decomposition (Miller & Johnson-Laird, 1976). The idea is that all the words in a semantic field share some primitive semantic concept, and it is the activation or suppression of that shared concept that affects the accessibility of the words sharing it.

Nominal semantic fields are frequently organized hierarchically and so are relatively simple to appreciate. Verbal semantic fields, however, tend to be more complex. For example, all the motion verbs -- "move," "come," "go," "bring," "rise," "fall," "walk," "run," "turn," and so on -- share a semantic primitive that might be glossed as "change location as a function of time." In a similar manner, verbs of possession -- "possess," "have," "own," "borrow," "buy," "sell," "find," and so on -- share a semantic primitive that has to do with rights of ownership.

Not all semantic primes nucleate semantic fields, however. There is a causative primitive that differentiates "rise" and "raise," "fall" and "fell," "die" and "kill," and so on, yet the causative verbs "raise," "fell," "kill" do not form a causative semantic field. Johnson-Laird and I distinguished two classes of semantic primitives: those (like motion) around which a semantic field can form, and those (like causation) used to differentiate concepts within a given field.

Although the nature of semantic primitives is a matter of considerable interest to anyone who proposes a semantic notation for writing the definitions that a language processing system will use, they have received relatively little attention from psychologists. Experimental psychologists have a strong tendency to concentrate on questions of function and process at the expense of questions of content. Perhaps their attempts to understand the processes of disambiguation will stimulate greater interest in these structural questions.

THE PROBLEM OF CONTEXT

The reason that lexical polysemy causes so little actual ambiguity is that, in actual use, context provides information that can be used to select the intended sense. Although contextual disambiguation is simple enough when people do it, it is not easy for a computer to do, even when the text is semantically well-formed. With semantically ill-formed input the problem is much worse.

I will illustrate the problem by describing some research we have been doing on vocabulary growth in school children. The results indicate that we need better ways to teach new words; with that need in mind I will return to the question of what we can reasonably expect from natural language interfaces.

Children's Use of Dictionaries

We have been looking at what happens when teachers send children to the dictionary to "look up a word and write a sentence using it." The results can be amusing: for example, Deese (1967) has reported on a 7th-grade teacher who told her class to look up "chaste" and use it in a sentence. Their sentences included: "The milk was chaste," "The plates were still chaste after much use," and "The amoeba is a chaste animal."

In order to understand what they were doing, you have to see the dictionary entry for "chaste":

CHASTE: 1. innocent of unlawful sexual intercourse. 2. celibate. 3. pure in thought and act, modest. 4. severely simple in design or execution, austere.

As Deese noted, each of the children's sentences is compatible with information provided by the dictionary that they had been told to consult.

You might think that Deese's observation was merely an amusing reflection of some quirk in the dictionary entry for "chaste," but that assumption would be quite wrong. Patti Gildea and I (Miller & Gildea, 1985) have confirmed Deese's observation many times over. We asked 5th and 6th grade children to look words up and to write sentences using them. As of this writing, our 10- and 11-year old friends have written a few thousand sentences for us, and we are still collecting them.

Our goal is to discover which kinds of mistakes are most frequent. In order to do this, we evaluate each sentence as we enter it into a data management system and, if something is wrong, we describe the mistake. By collecting our descriptions, we have made a first, tentative classification.

This project is still going on, so I can give only a preliminary report based on about 20% of our data. So far we have analyzed 457 sentences incorporating 22 target words: 12 are relatively common words that most of the children knew, and 10 are relatively rare words with which they were unfamiliar. The common words

were selected from the core vocabulary of words introduced by authors of 4th-grade basal readers; the rare words were selected from those introduced in 12th-grade readers (Taylor, Frackenpohl, & White, 1979). It is convenient to refer to them as the 4th-grade words and the 12th-grade words, respectively.

Errors were relatively frequent. Of the sentences classified so far, only 21% of those using 4th-grade words were sufficiently odd or unacceptable to indicate that the author did not have a good grasp on the meaning and use of the word, but 63% of the sentences using 12th-grade words were judged to be odd. Thus, the majority of the errors occurred with the 12th-grade words.

Table 2 shows our current classification. Note that the categories are not mutually exclusive: some ingenious youngsters are able to make two or even three mistakes in a single sentence.

Table 2
Classification of Sentences

Type of Sentence	4th-grade	12th-grade
No mistake	197(249)	76(208)
Selectional error	10	58
Wrong part of speech	4	41
Wrong preposition	4	24
Inappropriate topic	0	24
Used rhyming word	0	14
Inappropriate object	5	9
Wrong entry	4	9
Word not used	9	1
Object missing	5	3
Two senses confounded	4	3
No response	0	4
Not a word	0	3
Unacceptable idiom	3	0
Sentence not complete	3	0

Most of the descriptive phrases in Table 2 should be self-explanatory, but some examples may help. Skip the selectional errors; I shall say more about them in a moment.

Consider "Wrong part of speech": a student wrote "my hobby is listening to Duran Duran records, I have obtained an ACCRUE for it", thus using a verb as a noun. As an example of "Wrong preposition," consider the student who wrote: "Be very METICULOUS on your work." An example of "Inappropriate topic" is: "The train was TRANSITORY." An example of "Inappropriate object" is: "I was METICULOUS about falling off the cliff." Examples of "Used rhyming word" are "Did it ever ACCRUE to you that Maria T. always marks with a special pencil on my face?", "Did you evict that old TENET?", and "The man had a knee REPARATION."

Other categories were even less frequent, so return now to the most common type of mistake, the one labelled "Selectional error."

Violations of Selectional Preferences

The sentences that Deese reported illustrate selectional errors. Further examples can be taken from our data: "We had a branch ACCRUE on our plant," "I bought a battery that was TRANSITORY," "The rocket REPUDIATE off into the sky," "John is always so TENET to me."

It is unfair to call these sentences "errors" and to laugh at the children's mistakes. The students were doing their best to use the dictionary. If there was any mistake, it was made by adults who misunderstood the nature of the task that they had assigned.

Take the "accrue" sentence, for example. The definition that the students saw was:

ACCRUE: come as a growth or result: "Interest will accrue to you every year from money left in a savings bank. Ability to think will accrue to you from good habits of study."

We assume that the student read this definition looking for something she understood and found "come as a growth." She composed a sentence around this phrase: "We had a branch COME AS A GROWTH on our plant", then substituted "accrue" for it.

This strategy seems to account for the other examples. A familiar word is found in the definition, a sentence is composed around it, then the unfamiliar word is substituted for the familiar word. Some further evidence supports the claim that something like this strategy is being used. One intriguing clue is that sometimes the final substitution is not made: the written sentence contains the word selected from the definition but not the word that it defined. And, since substitution is not a simple mental operation for children, sometimes the selected word or phrase from the definition is actually written in the margin of the paper, alongside the requested sentence.

These are called selectional errors because they violate selectional preferences. For example, the girl who discovered that "stimulate" means "stir up" and so wrote, "Mrs. Jones stimulated the cake," violated the selectional preference that "stimulate" should take an animate object.

One reason these errors are so frequent is that dictionaries do not provide much information about selectional preferences. We think we know how to remedy that deficiency, but that is not what I want to discuss here. For the moment it suffices if you recognize that we have a plentiful supply of sentences containing violations of selectional preferences, and that the sentences are of some educational significance.

Intelligent Tutoring?

Now let me pose the following question. Could we use these sentences as a "bug catalog" in an intelligent tutoring system?

At the moment, intelligent tutoring systems (Sleeman & Brown, 1982) use many menus to obtain the student's answers to questions, and some people feel that this is actually an advantage. But I suspect that if we had a good language interface, one that understood natural language responses, it would soon replace the menus.

In any case, imagine an intelligent tutoring system that can handle natural language input. Imagine that the tutor asked children to write sentences containing words that they had just seen defined, recognized when a selectional error had occurred, then undertook to explain the mistake.

What would the intelligent tutor have to know in order to detect and correct a selectional error? Otherwise said, what more would it have to know than any language comprehender has to know?

The question is not rhetorical; I ask it because I would really like to know the answer. In my view, it poses something of a dilemma. The problem, as Yorick Wilks (1978) has pointed out, is that any simple rules of co-occurrence that we are likely to propose will, in real discourse, be violated as often as they are observed. (Not only do people often say one thing and mean another, but the prevalence of figurative and idiomatic language is consistently underestimated by theorists.) If we give the intelligent tutor strict rules in order to detect selectional errors like "Our car depletes gasoline," will it not also treat "Our car drinks gasoline" as an error? On the other hand, if the tutor accepted the latter, would it not also accept the former?

An even simpler dilemma, one often noted, is that a system that blocks such phrases as "colorless green ideas" will also block such sentences as "There are no colorless green ideas." If our tutor teaches children to avoid "stimulate the cake," will it also teach them to avoid "you can't stimulate a cake"?

When subtle semantic distinctions are at issue, it is customary to remark that a satisfactory language understanding system will have to know a great deal more that the linguistic values of words. It will have to know a great deal about the world, and about things that people presuppose without reflection. Such remarks are probably true, but they offer little guidance in getting the job done.

Since I have no better answer, I will simply agree that the lexical information available to any satisfactory language understanding system will have to be closely coordinated with the system's general information about the world. To pursue that idea would, of course, go beyond the lexical limits I have imposed here, but it does suggest that we will have to write our dictionary not once, but many times -- until we get it right.

So, while there is no principled obstacle to having large vocabularies in our natural language interfaces, there are still many problems to be solved. There is work here for everyone -- linguists, philosophers, and psychologists, as well as computer scientists -- and it is not abstract or impractical work. The answers we provide will shape important aspects of the information systems of the future.

References

Amsler, R. A. (1984) Machine-readable dictionaries. Annual Review of Information Science and Technology, 19, 161-209.

Becker, C. A. (1980) Semantic context effects in visual word recognition: An analysis of semantic strategies. Memory & Cognition, 8, 493-512.

Bolt, R. A. (1984) The Human Interface: Where People and Computers meet. Belmont, Calif.: Lifetime Learning.

Cullingford, R. E. (1985) Natural Language Processing: A Knowledge Engineering Approach. (Manuscript).

Deese, J. (1967) Meaning and change of meaning. American Psychologist, 22, 641-651.

Meyer, D. E., & Schvaneveldt, R. W. (1971) Faciliation in recognizing pairs of words: Evidence of a dependence between retrieval operations. Journal of Experimental Psychology, 90, 227-234.

Miller, G. A. (1977) Spontaneous Apprentices: Children and Language. New York: Seabury Press.

Miller, G. A. (1978) Semantic relations among words. In M. Halle, J. Bresnan, & G. A. Miller (eds.), Linguistic Theory and Psychological Reality. Cambridge, Mass.: MIT Press.

Miller, G. A., & Gildea, P. M. (1985) How to misread a dictionary. AILA Bulletin (in press).

Miller, G. A., & Johnson-Laird, P. N. (1976) Language and Perception. Cambridge, Mass.: Harvard University Press.

Procter, P. (ed.) (1978) Longman Dictionary of Contemporary English. Harlow, Essex: Longman.

chank, R. C. (1975) Conceptual Information Processing. Amsterdam: North-Holland.

Simpson, G. B. (1984) Lexical ambiguity and its role in models of word recognition. Psychological Bulletin, 96, 316-340.

Sleeman, D., & Brown, J. S. (eds.) (1982) Intelligent Tutoring Systems. New York: Academic Press.

Taylor, S. E., Frackenpohl, H., & White, C. E. (1979) A revised core vocabulary. In EDL Core Vocabularies in Reading, Mathematics, Science, and Social Studies. New York: McGraw-Hill.

Templin, M. C. (1957) Certain Language Skills in Children: Their Development and Interrelationships. Minneapolis: University of Minnesota Press.

Walker, D. E., & Amsler, R. A. (1984) The use of machine readable dictionaries in sublanguage analysis. In R. I. Kittredge (ed.), Workshop on Sublanguage Analysis. (Available from the authors at Bell Communications Research, 435 South Street, Morristown, NJ 07960.)

Wilks, Y. A. (1978) Making preferences more active. Artificial Intelligence, 11, 197-223.