A PROPOSAL FOR LEXICAL DISAMBIGUATION

George A. Miller Daniel A. Teibel
Princeton University Cognitive Science Laboratory
221 Nassau Street
Princeton, New Jersey 08542

ABSTRACT

A method of sense resolution is proposed that is based on WordNet, an on-line lexical database that incorporates semantic relations (synonymy, antonymy, hyponymy, meronymy, causal and troponymic entailment) as labeled pointers between word senses. With WordNet, it is easy to retrieve sets of semantically related words, a facility that will be used for sense resolution during text processing, as follows. When a word with multiple senses is encountered, one of two procedures will be followed. Either, (1) words related in meaning to the alternative senses of the polysemous word will be retrieved; new strings will be derived by substituting these related words into the context of the polysemous word; a large textual corpus will then be searched for these derived strings; and that sense will be chosen that corresponds to the derived string that is found most often in the corpus. Or, (2) the context of the polysemous word will be used as a key to search a large corpus; all words found to occur in that context will be noted; WordNet will then be used to estimate the semantic distance from those words to the alternative senses of the polysemous word; and that sense will be chosen that is closest in meaning to other words occurring in the same context. If successful, this procedure could have practical applications to problems of information retrieval, mechanical translation, intelligent tutoring systems, and elsewhere.

BACKGROUND

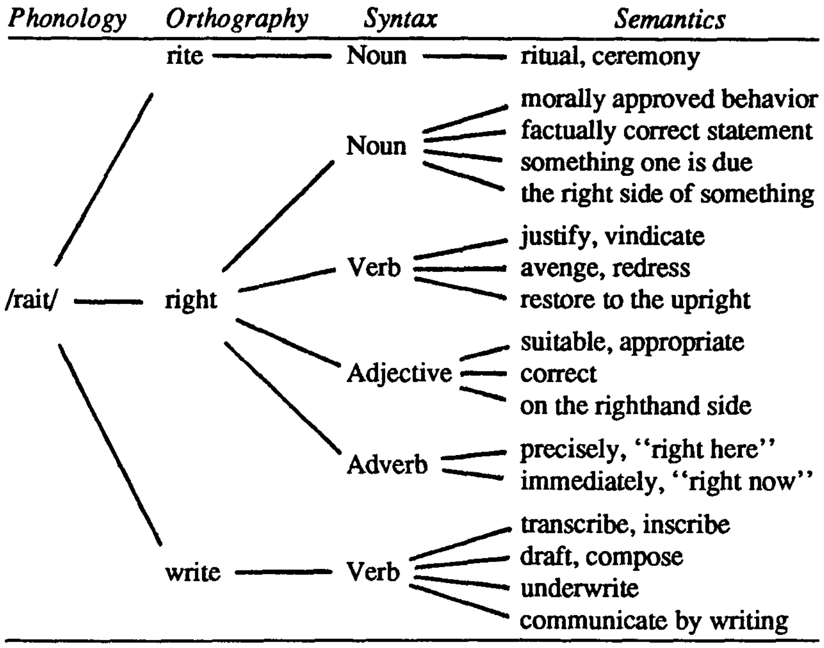
An example can set the problem. Suppose that an automatic transcription device were to recognize the string of phonemes /rait/ in the flow of speech and could correctly identify it as an English word; the device would still have to decide whether the word should be spelled *right*, *write*, or *rite*. And if the result were then sent to a language understanding system, there would be a further problem of deciding which sense of the word the speaker intended to communicate. These decisions, which are made rapidly and unconsciously by human listeners, are difficult to accomplish computationally. Ordinarily, anyone who reads and writes English will be able to listen to the context of a string like /rait/ and quickly decide which sense is appropriate. The task is so easy, in fact, that laymen unfamiliar with these matters find it hard to understand what the problem is. But computers have trouble using context to make such apparently simple lexical decisions. How those troubles might be overcome is the subject of this paper.

The process under consideration here is called "lexical disambiguation," although that terminology can be misleading. In the everyday use of linguistic communication, true ambiguity is remarkably rare. In the present context, however, "ambiguity" has taken on a special meaning that derives, apparently, from the claim by Katz and Fodor (1963) that semantics, like syntax, should be restricted to sentences, "without reference to information about settings" (p. 174). Many sentences are indeed ambiguous when viewed in a contextual vacuum. More to the point, as Katz and Fodor emphasized, most words, when taken in isolation, are ambiguous in just this sense; they convey different meanings when used in different linguistic settings. Hence, lexical disambiguation is the process (either psychological or computational) that reduces this putative ambiguity—that results in the selection of the appropriate sense of a polysemous word. "Sense resolution" might be a better term, but by now "disambiguation" is firmly established in the technical literature.

Much attention has been given to lexical disambiguation by students of language understanding. In an excellent survey, Hirst (1987) distinguishes three types of lexical ambiguity: categorical, homonymous, and polysemous. A word is categorically ambiguous if it can be used in different syntactic categories; *right*, for example, can be used as a noun, a verb, an adjective, or an adverb. A word is a homonym if it has two clearly different senses; as an adjective, for example, *right* can mean the opposite of *wrong* or the opposite of *left*. A word is polysemous if it has senses that are different but closely related; as a noun, for example, *right* can mean something that is morally approved, or something that is factually correct, or something that is due one. So defined, the distinction between homonymy and polysemy becomes a matter of degree that is often difficult to draw; some lexicographers have tried to draw it on etymological grounds. For the present discussion, however, the distinction will be ignored; homonymy and polysemy will be referred to together simply as polysemy.

Categorical ambiguity, however, is of a different kind and is resolved in a different way. For the purposes of the present paper, it will be assumed that only content words are at issue, and that the syntactic category of all content words in the text that is under study can be determined automatically (Church, 1988; DeRose, 1988). The problem is simply to decide which sense of a content word—noun, verb, adjective, or adverb—is appropriate in a given linguistic context. It will also be assumed that sense resolution for individual words can be accomplished on the basis of information about the immediate linguistic context. It should be noted that this is an important simplification. Inferences based on knowledge of the world are frequently required to resolve uncertainties about a speaker's intentions in uttering a particular sentence, and endowing computers with such general knowledge in a usuable form is a particularly formidable problem. Sense resolution for individual words, however, promises to be more manageable.

Illustrating Multiple Senses of an English Word



Hirst (1987) has reviewed various attempts to program computers to use linguistic contexts in order to perform lexical disambiguation; that information need not be repeated here. It should be noted, however, that there are two contrasting ways to think about linguistic contexts, one based on co-occurrence and the other on substitutability (Charles and Miller, 1989; Miller and Charles, 1991), usually referred to (Jenkins, 1954) as the syntagmatic and paradigmatic views. The co-occurrence or syntagmatic approach holds the target word constant and compares the contexts in which it can appear; the substitutability or paradigmatic approach holds the context constant and compares the words that can appear in it. According to the co-occurrence approach, associations are formed between a word and the other words that occur with it in the same phrases and sentences; most psycholinguists assume that syntagmatic word associations are a consequence of temporal and spatial contiguity. Many attempts to automate lexical disambiguation have exploited these co-occurrence associations. Lesk (1986) provides an elegantly simple example: each sense of the polysemous word is retrieved from an on-line dictionary and compared with the target sentence; the sense is chosen that has the most words in common with the target sentence. According to the substitutability view, on the other hand, associations are formed between words that can be substituted into similar contexts; most psycholinguists assume that paradigmatic word associations between words are mediated by their common contexts. Syntactic categories of words, for example, must be learned on the basis of their inter-substitutability; Miller and Charles (1991) have argued that semantic similarities between words are also learned on the basis of inter-substitutability.

The present proposal is an attempt to exploit paradigmatic associations for the purpose of lexical disambiguation. That is to say, it is proposed to use the substitutability of semantically similar words in order to determine which sense of a polysemous word is appropriate. In order to explain where the semantically similar words might come from, however, it is necessary to describe the lexical database that is a central component of the present proposal for lexical disambiguation.

WordNet

Standard alphabetical procedures for organizing lexical information put together words that are spelled alike and scatter words with related meanings haphazardly through the list. WordNet (Miller, 1990) is an attempt to use computers in order to achieve a more efficient organization of the lexicon of English. Inasmuch as it instantiates hypotheses based on results of psycholinguistic research, it can be said to be a dictionary based on psycholinguistic principles. One obvious difference from a conventional on-line dictionary is that WordNet divides the lexicon into four syntactic categories: nouns, verbs, modifiers, and function words. In fact, WordNet contains only nouns, verbs, and adjectives. Adverbs are omitted on the assumption that most of them duplicate adjectives; the relatively small set of English function words is omitted on the assumption that they are stored separately as part of the syntactic component. The most ambitious feature, however, is the attempt to organize lexical information in terms of word meanings, rather than word forms. In that respect, WordNet resembles a thesaurus. It is not merely an on-line thesaurus, however. In order to appreciate what more has been attempted, it is necessary to understand the basic design.

Lexical semantics begins with a recognition that a word is a conventional association between a lexicalized concept and an utterance that plays a syntactic role. The basic structure of any lexicon is a many:many mapping between word forms and word senses (Miller, 1986), with syntactic category as a parameter. When a particular word form can be used to express two or more word senses it is said to be polysemous; when a particular word sense can be expressed by two or more word forms they are said to be synonymous (relative to a context). Initially, WordNet was to be concerned solely with the relations between word senses, but as the work proceeded it became increasingly clear that questions of relations between word forms could not be ignored. For lexical disambiguation, however, word meanings are crucial, so this description will focus on semantic relations.

How word senses are to be represented is a central question for any theory of lexical semantics. In WordNet, a lexicalized concept is represented by simply listing the word forms that can (in an appropriate context) be used to express it: {W_1, W_2, . . . }. (The curly brackets are used to surround sets of synonyms, or *synsets.*) For example, *board* can signify either a piece of lumber or a group of people assembled for some purpose; these two senses can be represented by the synsets {*board, plank*} and {*board, committee*}. These synsets do not explain what the concepts are; they serve merely to signal that two different concepts exist. People who know English are assumed to have already acquired the concepts and are expected to recognize them from the words listed in the synsets.

The mapping between word forms and word senses, therefore, can be represented as a mapping between written words and synsets. Since English is rich in synonyms, synsets are often sufficient for differentiation, but sometimes an appropriate synonym is not available. In that case, the lexicalized concept can be represented by a short gloss, e.g., {*board,* (a person's meals, provided regularly for money)}. The gloss is not intended for use in constructing a new lexical concept, and differs from a synonym in that it is not used to gain access to stored information. Its purpose is simply to enable users to distinguish this sense from others with which it could be confused.

WordNet is organized by semantic relations. A great variety of semantic relations could be defined, of course, but this work was limited not only to relations that lay persons can appreciate without advanced training in linguistics, but also to relations that have broad application throughout the lexicon. In that way it was hoped to capture the gross semantic structure of the English lexicon, even though particular semantic domains (e.g., kin terms, color terms) may not be optimally analyzed.

Since a semantic relation is a relation between meanings, and since meanings are here represented by synsets, it is natural to think of semantic relations as labeled pointers between synsets. In the case of synonymy and antonymy, however, the semantic relation is a relation between words.

Synonymy: The most important semantic relation in WordNet is synonymy, since it allows the formation of synsets to represent senses. According to one definition (usually attributed to Leibniz) two expressions are synonymous if the substitution of one for the other never changes the truth value of a statement in which the substitution is made. By that definition, true synonyms are rare in natural languages. A weakened version of the definition would make synonymy relative to a context: two expressions are synonymous in a context C if the substitution of one for the other in C does not alter the truth value. For example, the substitution of *plank* for *board* will seldom alter the truth value in carpentry contexts, although in other contexts that substitution might be totally inappropriate. Note that this definition of synonymy in terms of substitutability makes it necessary to partition the lexicon into nouns, adjectives, and verbs. That is to say, if concepts are represented by synsets, and if synonyms must be inter-substitutable, then words in different syntactic categories cannot form synsets because they are not inter-substitutable.

Antonymy: Another familiar semantic relation is antonymy. Like synonymy, antonymy is a semantic relation between words, not between concepts. For example, the meanings {*rise, ascend*} and {*fall, descend*} may be conceptual opposites, but they are not antonyms; *rise/fall* are antonyms, and *ascend/descend* are antonyms, but most people hesitate and look thoughtful when asked whether *rise* and *descend,* or *fall* and *ascend,* are antonyms. Antonymy provides the central organizing relation for adjectives: every predicable adjective either has a direct antonym or is similar to another adjective that has a direct antonym. *Moist,* for example, does not have a direct antonym, but it is similar to *wet,* which has the antonym *dry*; thus, *dry* is an indirect antonym of *moist.*

Hyponymy: Hyponymy is a semantic relation between meanings: e.g., {*maple*} is a hyponym of {*tree*}, and {*tree*} is a hyponym of {*plant*}. Considerable attention has been devoted to hyponymy/hypernymy (variously called subordination/superordination, subset/superset, or the ISA relation). Hyponymy is transitive and asymmetrical, and, since there is normally a single superordinate, it generates a hierarchical semantic structure, or tree. Such hierarchical representations are widely used in information retrieval systems, where they are known as inheritance systems (Touretsky, 1986); a hyponym inherits all of the features of is superordinates. Hyponymy provides the central organizing principle for nouns.

Meronymy: The part/whole (or HASA) relation is known to lexical semanticists as meronymy/holonymy. One concept *x* is a meronym of another concept *y* if native speakers accept such constructions as *An x is a part of y* or *y has x as a part.* If *x* is a meronym of *y*, then it is also a meronym of all hyponyms of *y*.

Entailment: A variety of entailment relations hold between verbs. For example, the semantic relation between *kill* and *die* is one of causal entailment; to kill is to cause to die. Similarly, the semantic relation between *march* and *walk* is troponymy, an entailment of manner; to march is to walk in a certain manner. Other types of entailment hold between *marry* and *divorce*; a divorce entails a prior marriage. These entailments, along with synonymy and antonymy, provide the central organizing principles for the verb lexicon.

It should be obvious that, given this semantic organization of the lexical database, it is a simple matter to retrieve sets of words that have similar senses. The next step is to consider how such related words can be used for lexical disambiguation.

THE PROPOSED SYSTEM

It is assumed that a grammatical text is to be processed, and that the processor is expected to use the textual context to determine the appropriate sense of each successive content word. Then, in brief outline, the present proposal envisions a processor that will perform three operations:

(1) Take a content word from the text and look it up in the lexical database; if a single sense is found, the problem is solved. If more than one sense is found, continue.

(2) Determine the syntactic category of each sense. If a single category is involved, go to operation three. If more than one syntactic category is found, use a "parts" program to determine the appropriate category. If the word has only one sense as a member of that category, the problem is solved. If the word has more than one sense in the appropriate syntactic category, continue.

(3) Determine which sense of the polysemous word is appropriate to the text. If the word is a noun, determine which sense can serve as an argument of the verb, or can be modified by an accompanying adjective. If the word is verb or adjective, determine which sense can be combined with an accompanying noun phrase.

The final operation is the critical step, of course, but before describing how it might be implemented, a simplified example will help to make the central idea clear. Suppose the processor encounters the sentence, *the baby is in the pen*, and tries to assign the appropriate sense to the noun *pen*. It would first generalize the given context (e.g., with respect to number and tense), then find words that are semantically related to the various senses of *pen* and substitute them into the generalized context. It would then undertake a comparison of:

(a/the baby is/was)/(the/some babies are/were) in a/the:
(a) *fountain pen/pencil/quill/crayon/stylus*
(b) *sty/coop/cage/fold/pound*
(c) *playpen/playroom/nursery*
(d) *prison/penitentiary/jail/brig/dungeon*
(e) *swan/cygnet/goose/duck/owl*

In order to decide that one of these is acceptable and the others are unlikely, the processor might search an extensive corpus for strings of the form "(a/the baby is/was)/(the babies are/were) in the *X*," where *X* is one of the closely related words listed above. If the *playpen/playroom/nursery* expressions significantly outnumber the others, the conventionally correct choice can be made. In other words, the processor will interrogate a corpus much the way a linguist might ask a native informant: "Can you say this in your language?"

That is the basic strategy. Words related in meaning to the different senses of the polysemous word will be retrieved; new expressions will be derived by substituting these related words into the generalized context of the polysemous word; a large textual corpus will then be searched for these derived expressions; that sense will be chosen that corresponds to the derived expression that is found most often in the corpus. (Alternatively, all contexts of the semantically related words could be collected and their similarity to the target context could be estimated.)

We assume that the similarity of this strategy to the theory of spreading activation (Quillian, 1968, 1969) is obvious. Of course, in order even to approach the best possible implementation, a variety of possibilities will have to be explored. For example, how much context should be preserved? Too short, and it will not discriminate between different senses; too long and no instances will be found in the corpus. Should the grammatical integrity of the contexts be preserved? Or, again, how large a corpus will be required? Too small, and no instances will be found; too large and the system will be unacceptably large or the response unacceptably slow. Fortunately, most of the polysemous words occur relatively frequently in everyday usage, so a corpus of several million words should be adequate. Or, still again, how closely related should the semantically related words be? Can superordinate terms be substituted? How far can the contexts be generalized? Experience should quickly guide the choice of sensible answers.

As described so far, the procssor begins with WordNet in order to find semantically related words that can be searched for in a corpus. Obviously, it could all be done in the reverse order. That is to say, the processor could begin by searching the corpus for the given generalized context. In the above example, it might search for "(a/the baby is/was)/(the babies are/were) in the *Y*," where *Y* is any word at all. Then, given the set of *Y* words, WordNet could be used to estimate the semantic distance from these words to the alternative senses of the polysemous word. A similarity metric could easily be constructed by simply counting the number of pointers between terms. That sense would be chosen that was closest in meaning to the other words that were found to occur in the same context.

Whether WordNet is used to provide related words or to measure semantic similarity, a major component of the present proposal is the search of a large textual corpus. Since the corpus would not need to be continually updated, it should be practical to develop an inverted index, i.e., to divide the corpus into sentence items that can be keyed by the content words in WordNet, then to compute hash codes and write inverted files (Lesk, 1978). In this way, a small file of relevant sentences could be rapidly assembled for more careful examination, so the whole process could be conducted on-line. Even if response times were satisfactorily short, however, one feels that once a particular context has been used to disambiguate a polysemous word, it should never have to be done again. That thought opens up possibilities for enlarging WordNet that we will not speculate about at the present time.

SOME OBVIOUS APPLICATIONS

Several practical applications could result from a reliable lexical disambiguation device. The fact that people see concepts where computers see strings of characters is a major obstacle to human-machine interaction.

Consider this situation. A young student who is reading an assignment encounters an unfamiliar word. When a dictionary is consulted it turns out that the word has several senses. The student reconsiders the original context, testing each definitional gloss in turn, and eventually chooses a best fit. It is a slow process and a serious interruption of the student's task of understanding the text. Now compare this alternative. A computer is presenting a reading assignment to the same student when an unfamiliar word appears. The student points to the word and the computer, which is able solve the polysemy problem, presents to the student only the meaning that is appropriate in the given context—as if a responsive teacher were sitting at the student's side. The desired information is presented rapidly and the real task of understanding is not interrupted.

Or think of having a lexical disambiguator in your word processing system. As you write, it could flag for you every word in your text that it could not disambiguate on the basis of the context you have provided. It might even suggest alternative wordings.

The application to mechanical translation is also obvious. A polysemous word in the source language must be disambiguated before an appropriate word in the target language can be selected. The feasibility of multilingual WordNets has not been explored.

Finally, consider the importance of disambiguation for information retrieval systems. If, say, you were a radar engineer looking for articles about antennas and you were to ask an information retrieval system for every article it had with *antenna* in the title or abstract, you might receive unwanted articles about insects and crustaceans—the so-called problem of false drops. So you revise your descriptor to, say, *metal antenna* and try again. Now you have eliminated the animals, but you have also eliminated articles about metal antennas that did not bother to include the word *metal* in the title or abstract—the so-called problem of misses. False drops and misses are the Scylla and Charybdis of information retrieval; anything that reduces one tends to increase the other. But note that a lexical disambiguator could increase the probability of selecting only those titles and abstracts in which the desired sense was appropriate; the efficiency of information retrieval would be significantly increased.

In short, a variety of practical advances could be implemented if it were possible to solve the problem of lexical ambiguity in some tidy and reliable way. The problem lies at the heart of the process of turning word forms into word meanings. But the very reason lexical disambiguation is important is also the reason that it is difficult.

REFERENCES

Charles, W. G., and Miller, G. A. "Contexts of antonymous adjectives," *Applied Psycholinguistics*, Vol. 10, 1989, pp. 357-375.

Church, K. "A stochastic parts program and noun phrase parser for unrestricted text," *Second Conference on Applied Natural Language Processing*, Austin, Texas, 1988.

DeRose, S., "Grammatical category disambiguation by statistical organization," *Computational Linguistics*, Vol. 14, 1988, pp. 31-39.

Hirst, G., *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge, 1987.

Jenkins, J. J., "Transitional organization: Association techniques." In C. Osgood and T. A. Sebeok (Eds.), *Psycholinguistics: A Survey of Theory and Research Problems*. Supplement, *Journal of Abnormal and Social Psychology*, Vol. 52, 1954, pp. 112-118.

Katz, J. J., and Fodor, J. A. "The structure of a semantic theory," *Language*, Vol. 39, 1963, pp. 170-210.

Lesk, M. E. "Some applications of inverted indexes on the Unix system," *Unix Programmer's Manual*, Vol. 2a, Bell Laboratories, Murray Hill, NJ, 1978.

Lesk, M. E. "Automatic sense discrimination: How to tell a pine cone from an ice cream cone," manuscript, 1986.

Miller, G. A. "Dictionaries in the mind," *Language and Cognitive Processes*, Vol. 1, 1986, 171-185.

Miller, G. A. (Ed.) Five papers on WordNet, *International Journal of Lexicography*, Vol. 3, No. 4, 1990, 235-312.

Miller, G. A., and Charles, W. G. "Contextual correlates of semantic similarity," *Language and Cognitive Processes*, Vol. 6, 1991.

Quillian, M. R. "Semantic memory." In Minsky, M. L. (Ed.) *Semantic Information Processing*. MIT Press, Cambridge, MA, 1968.

Quillian, M. R. "The teachable language comprehender: A simulation program and theory of language." *Communications of the ACM*, Vol. 12, 1969, 459-476.

Touretzky, D. S. *The Mathematics of Inheritance Systems*. Morgan Kaufman, Los Altos, California, 1986.