NATURAL LANGUAGE UNDERSTANDING

Avron Barr
Department of Computer Science
Stanford University
Palo Alto, CA 94305

This is an excerpt from the Handbook of Artificial Intelligence, a compendium of hundreds of articles about AI ideas, techniques, and programs being prepared at Stanford University by AI researchers and students from across the country. In addition to articles describing the specifics of various AI programming methods, the Handbook contains dozens of overview articles like this one, which attempt to give historical and scientific perspective to work in the different areas of AI research.

This article is from the Handbook chapter on Natural Language Understanding. Cross-references to other articles in the handbook have been removed—terms discussed in more detail elsewhere are italicized. Many people have contributed to this chapter, including especially Anne Gardner, James Davidson, and Terry Winograd. Avron Barr and Edward A. Feigenbaum are the Handbook's general editors.

The most common way that people communicate is by speaking or writing in one of the "natural" languages, like English, French, or Chinese. Computer programming languages, on the other hand, seem awkward to humans. These "artificial" languages are designed so that sentences have a rigid format, or *syntax*, making it easier for compilers to *parse* a program and convert it into the proper sequence of computer instructions. Besides being structurally simpler than natural languages, programming languages can express easily only those concepts that are important in programming: "Do this, then do that," "See whether such and such is true," etc. The things that can be meant by expressions in a language are referred to as the *semantics* of the language.

The research described in this chapter of the Handbook concerns the development of computer programs that try to deal with the full range of meaning of languages like English. If computers could understand what people mean when typing (or speaking) English sentences, they would be easier to use and would fit more naturally into people's lives. Furthermore, artificial intelligence (AI) researchers hope that learning how to build computers that can communicate as people do will extend our understanding of the nature of language and of the mind.

So far, programs have been written that are quite successful at processing somewhat constrained input: the user is limited in either the structural variation of his sentences (syntax constrained by an artificial *grammar*) or in the number of things he can "mean" (in domains with constrained semantics). Some of these systems are adequate and are available commercially. But the fluent use of language as humans use it is still elusive, and natural language (NL) understanding is and active area of research in AI.

This article presents a brief sketch of the history of natural language processing research and an idea of the current state

of the art. The other articles in the NL chapter of the Handbook include a historical sketch of *machine translation* from one language to another, which was the subject of the very earliest ideas about processing language with computers; technical articles on some of the grammars and parsing techniques that AI researchers have used in their programs; and an article on *text generation*, the creation of sentences by the program. Finally, there are several articles describing the NL programs themselves: the early systems of the 1960s and the major research projects of the last decade, including Wilks's machine translation system, Winograd's SHRDLU, Woods's LUNAR, Schank's MARGIE, SAM, and PAM, and Hendrix's LIFER.

Two other chapters of the Handbook are especially relevant to NL research. *Speech understanding* research attempts to build computer interfaces that understand spoken language. In the 1970s, speech and natural language understanding research were often closely linked. Increasingly inseparable from NL research is the study of *knowledge representation*, because AI researchers have come to believe that a very large amount of knowledge about the world is needed to understand even simple dialogues. AI research in the representation of knowledge explores ways of making this *world knowledge* accessible to the computer program by building "representational" data structures in the machine's memory.

Early History

Research in *computational linguistics*, the use of computers in the study of language, started soon after computers became available in the 1940s (Bott 1970). The machine's ability to manipulate symbols was readily used on written text to compile word indexes (lists of word occurrences) and *concordances* (indexes including a line of context for each occurrence). Such surface-level machine processing of text was of some value in linguistic research, but it soon became apparent that the computer could perform much more powerful linguistic functions than merely counting and rearranging data.

In 1949, Warren Weaver proposed that computers might be useful for "the solution of world-wide translation problems." The resulting research effort, called *machine translation*, attempted to simulate with a computer the presumed functions of a human translator: looking up each word in a bilingual dictionary; choosing an equivalent word in the output language; and, after processing each sentence in this way, arranging the resulting string of words to fit the output language's word order.

Despite the attractive simplicity of the idea, many unforeseen problems arose, both in selecting appropriate word equivalences and in arranging them to produce a sentence in the output language. The concept of translating by replacing words with their equivalents and then fixing up the word order was abandoned. In its place, eventually, "understanding" became the focus of AI research in language—if the machine could actually understand the "meaning" of a sentence, it could presumably translate it into another language, answer questions about it, or paraphrase it. But the nature of understanding is itself a difficult problem. New AI approaches to natural language processing were influenced by many scientific developments of the 1960s, including "high-level" programming languages and *list processing*, vastly expanded computer power and memory capacity, and Chomsky's breakthroughs in linguistic theory.

In the 1960s, researchers developed a new group of computer programs, attempting to deal with some of the issues that had thwarted machine translation efforts. These early natural language programs mark the beginning of artificial intelligence work in understanding language. They begin to view human language as a complex cognitive ability involving knowledge of different kinds: the structure of sentences, the meaning of words, a model of the listener, the rules of conversation, and an extensive shared body of general information about the world. The current AI approach has been to model human language as a *knowledge-based system* for processing communications and to create computer programs that serve as working models of such a system.

AI researchers in natural language processing expect their work to lead both to the development of practical, useful language understanding systems and to a better understanding of language and the nature of intelligence. The computer, like the human mind, has the ability to manipulate symbols in complex processes, including processes that involve decision making based on stored knowledge. It is an assumption of the field that the human use of language is a cognitive process of this sort. By developing and testing computer-based models of language processing that approximate human performance, researchers hope to understand better how human language works.

Approaches to NL Processing

Natural language research projects have had diverse emphases and have used diverse methods, making their classification difficult. One coherent scheme, borrowed from Winograd (1972), groups natural language programs according to how they represent and use knowledge of thier subject matter. On this basis, natural language programs can be divided into four historical categories.

The earliest natural language programs sought to achieve only limited results in specific, constrained domains. These programs, like Green's BASEBALL, Lindsay's SAD-SAM, Bobrow's STUDENT, and Weizenbaum's ELIZA, used *ad hoc* data structures to store facts about a limited domain. Input sentences were restricted to simple declarative and interrogative forms and were scanned by the programs for predeclared key words or patterns which indicated known objects and relationships. Domain-specific rules, called *heuristics*, were used to derive the required response from the key words in the sentence and the knowledge in the database. Because their domains of discourse were so restricted, these early systems were able to ignore many of the complexities of language and achieve sometimes impressive question-answering results. (Weizenbam, 1976, argues that to the

extent the results were impressive, they were also misleading.)

Another early approach to NL processing was tried in PROTOSYNTHEX-I (Simmons et al. 1966) and Semantic Memory (Quillian 1968). These systems essentially stored a representation of the text itself in their databases, using a variety of clever indexing schemes to retrieve material containing specific words or phrases. In this *text-based* approach, the systems were not tied by their construction to a specific domain, since the textual database could cover any subject. However, they were still severely restricted in the sense that they could only respond with material that had been pre-stored explicitly. Though more general than their predecessors, these programs still failed to notice even obvious implications of the sentences in the database because they did not deal with the *meaning* of the English language input—they had no *deductive* powers.

To approach the problem of how to characterize and use the meaning of sentences, a third group of programs was developed during the mid-1960s. In these *limited-logic systems*, including Raphael's SIR, Quillian's TLC, Thompson's DEACON, and Kellogg's CONVERSE, the information in the database was stored in some formal notation and mechanisms were provided for translating input sentences into this internal form (*semantic analysis*). The formal notation was an attempt to liberate the informational content of the input from the structure of English. The overall goal of these systems was to perform inferences on the database in order to find answers to questions that were not stored explicitly in the database. For instance, if a system has been told that "Fido is a collie," and that "All collies are dogs," then it should be able to answer the question, "Is Fido a dog?" The systems of this period were "limited" in the sense that the deductions they could make were only a subset of the full range of logical inferences used in ordinary conversation.

The fourth group of natural language understanding programs might be called *knowledge-based systems*; their development if closely intertwined with AI research on the *representation of knowledge*. The programs use a large amount of information about the domain under discussion to help understand sentences—knowledge that is stored within the program using a representation scheme like *first order logic, procedural semantics, semantic networks,* or *frames*. But before discussing these knowledge-based systems of the 1970s, we should first mention an important development in the study of language during the preceding decade which strongly influenced their design.

Grammars and Parsing

A *grammar* of a language is a scheme for specifying the sentences allowed in the language, indicating the syntactic rules for combining words into well-formed phrases and clauses. The theory of *generative grammar* introduced by Chomsky (1957) radically influenced all linguistic research, including AI work in computational linguistics. In natural language processing programs, the grammar is used in *parsing* to "pick apart" the sentences in the input to the program to help determine their meaning and thus an appropriate response. Several very different types of grammars have been used in NL programs, including *phrase structure grammars, transformational grammars, case grammars,* and *semantic grammars.*

Parsing is the "delinearization" of linguistic input, that is, the use of grammatical rules and other sources of knowledge to determine the functions of the words in the input sentence in order to create a data structure, like a *derivation tree*, that can be used to get at the "meaning" of the sentence. All natural language processing computer systems contain a parsing component of some sort, but those of the early NL progrmas were based on key words expected in the input or were constrained to quite limited phrase structures. The practical application of grammars to the full range of natural language has proven difficult.

The design of a parser is a complex problem, both in theory and implementation. The first part of the design concerns the specification of the grammar to be used. The rest of the parsing system is concerned with the method of *use* of the grammar, that is, the manner in which strings of words are matched against patterns of the grammar. These considerations run into many of the general questions of computer science and artificial intelligence concerning process control and manipulation of representational data structures (See, for example, Marcus's rescent PARSIFAL system).

Knowledge-based NL Systems

In the early 1970s, two systems were built that attempted to deal with both syntactic and semantic aspects of language processing in a comprehensive way. William Woods's LUNAR program answered questions about the samples of rock brought back from the moon, using a large database provided by the National Aeronautics and Space Agency. It was one of the first programs to attack the problems of English grammar, using an *augmented transition network* parser. It used a notion of *procedural semantics* in which queries were first converted in a systematic way into a "program" to be executed by the retrieval component. Terry Winograd's SHRDLU system carried on a dialogue with a user in which the sytem simulated a robot manipulating a set of simple objects on a table top. The naturalness of the dialogue, as well as SHRDLU's apparent reasoning ability, made it particularly influential in the development of AI ideas about natural language processing. These two systems integrate syntactic and semantic analysis with a body of world knowledge about a limited domain, enabling them to deal with more sophisticated aspects of language and discourse then had previously been possible.

Central to these two systems is the idea of representing knowledge about the world as procedures within the system. The meanings of words and sentences were expressed as programs in a computer language, and the execution of these programs corresponded to reasoning from the meanings. Direct *procedural representations* are often the most straightforward way to implement the specific reasoning steps

needed for a natural language system. Most of the actual working systems that have been developed have made heavy use of specialized procedural representations, to fill in those places where the more *declarative* representation schemes—those where the "knowledge" is encoded in passive data structures that are interpreted by other procedures—are insufficient. (The *procedural/declarative controversy* was at one time an important focus in the development of AI. See Winograd, 1975.)

Perhaps the most influential declarative representation schemes were *logic* and *semantic networks*. Semantic networks were first proposed by Quilllian (1968) as a model for human associative memory. They used the concepts of graph theory, representing words and meanings as a set of linked nodes implemented as data structures in the computer program. By using a systematic set of link types, Quilliam was able to program simple operations (such as following chains of links) that corresponded to drawing inferences. The advantage of semantic networks over standard logic is that some selected set of the possible inferences can be made in a specialized and efficient way. If these correspond to the inferences that people make naturally, then the system will be able to do a more natural sort of reasoning than can be easily achieved using formal logical deduction. Semantic networks have been the basis for representation of the knowledge in a number of systems, including most of the *speech understanding* systems (Lea 1979).

Case representations extend the basic notions of semantic nets with the idea of a *case frame*, a cluster of the properties of an object or event into a single concept. There have been a large number of variations on this notion, some of which remain close to the linguistic forms. Others, such as *conceptual dependency*, are based on the notion of *semantic primitives*, the construction of all semantic notions from a small set of "primitive" concepts. Conceptual dependency representation was developed by Roger Schank and his colleagues and used in their NL systems, MARGIE and SAM (Schank 1975; Schank and Abelson 1977).

As with semantic networks, the advantage of case representations lies in their focus on grouping relevant sets of relationships into single data structures. The idea of clustering structures in a coherent and efficient way has been carried much further in representation schemes based on the notion of a *frame* (Minsky 1975). Where case representations deal primarily with single sentences or acts, frames are applied to whole situations, complex objects, or series of events. In analyzing a sentence, narrative, or dialogue, a frame-based language understanding system tries to match the input to the prototypes of objects and events in its domain stored in its database.

For example, Schank's SAM system makes use of frame-like data structures called *scripts*, which represent stereotyped sequences of events, to understand simple stories. It assumes that the events being described will fit (roughly) into one of the scripts in its knowledge base, which it then uses to fill in missing pieces in the story. The GUS system (Bobrow et al. 1977) is an experimental, frame-based travel consultant, engaging in dialogue to help a person schedule an air trip.

The important common element in all of these systems is that the existence of prototyped frames makes it possible to use *expectations* about the usual properties of known concepts and about what typically happens in a variety of familiar situations to help understand language. When a sentence or phrase is input that is ambiguous or underspecified, it can be compared to a description of what would be expected based on the prototype. Assumptions can be made about what was meant, if there is a plausible fit to the expectation. This *expectation-driven* processing seems to be an important aspect of the human use of language, where incomplete or ungrammatical sentences can be understood in appropriate contexts.

Investigation of script- and frame-based systems is the most active area of AI research in natural language understanding at the present time. Recent systems expand the domain of expectations used in processing language beyond those involving typical objects and events to include those based on how people use plans to achieve goals (Schank and Abelson 1977; Wilensky 1979) and on the rules people seem to follow in a dialogue (Cohen and Perrault 1979; Grosz 1979; Kaplan 1978; Robinson et al. 1980). The current state of the art in operational (non-experimental) NL systems is exemplified by ROBOT (Harris 1979), LIFER (Hendrix 1977), and PHLIQA1 (Landsbergen 1976).

References

General discussions of natural language processing research in AI can be found in Boden (1977), Charniak and Wilks (1976), Schank and Abelson (1977), and Winograd (forthcoming). Waltz (1977) contains more than fifty brief summaries of recent projects and systems. In addition, many historically important NL systems are described in the collections of articles edited by Feigenbaum and Feldman (1963), Minsky (1968), and Rustin (1973). Bobrow and Collins (1975), COLING (1976), Schank and Nash-Webber (1975), and Waltz (1978) are proceedings of recent conferences about current AI work in natural language. Related AI work on speech understanding is described in the collection of articles edited by Lea (1979). Many of the references below refer to the biannual International Joint Conference on Artificial Intelligence (IJCAI).

(1) Bobrow, D. G. 1968. Natural language input for a computer problem-solving system. In Minsky, pp. 146-226.

(2) Bobrow, D. G., and Collins, A. (Eds.) 1975. *Representation and Understanding: Studies in Cognitive Science*. New York: Academic Press.

(3) Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., and Winograd, T. 1977. GUS, a frame-driven dialogue system. *Artificial Intelligence*, 8, pp. 155-173.

(4) Boden, M. 1977. *Artificial Intelligence and Natural Man*. New York: Basic Books.

(5) Bott, M. F. 1970. Computational linguistics. In J. Lyons (Ed.), *New Horizons in Linguistics*. Harmondsworth: Penguin Books, pp. 215-228.

(6) Charniak, E., and Wilks, Y. 1976. *Computational Semantics: An Introduction to Artificial Intelligence and Natural Language Comprehension*. Amsterdam: North-Holland.

(7) Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton and Co.

(8) Cohen, P. R., and Perrault, C. R. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3, pp. 177-212.

(9) COLING. 1976. Preprints of the 6th International Conference on Computational Linguistics. Ottawa, Ontario, Canada.

(10) Feigenbaum, E. A., and Feldman, J. (Eds.) 1963. *Computers and Thought*. New York: McGraw-Hill.

(11) Green, B. F., Jr., Wolf, A. K., Chomsky, C., and Laughery, K. 1963. BASEBALL: An automatic question answerer. In Feigenbaum and Feldman, pp. 207-216.

(12) Grosz, B. J. 1979. Utterance and objective, issues in natural language communication. *IJCAI 6*, pp. 1067-1076.

(13) Harris, L. 1979. Experience with ROBOT in twelve commercial natural language data base query applications. *IJCAI 6*, pp. 365-368.

(14) Hendrix, G. G. 1977. Human engineering for applied natural language processing. *IJCAI 5*, pp. 183-191.

(15) Kaplan, S. J. 1979. Cooperative responses from a portable natural language data base query system. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.

(16) Kellogg, C. 1968. A natural language compiler for on-line data management. *Proceedings of the AFIPS Fall Joint Computer Conference*, 33. Washington: Thompson Book Co., pp. 473-492.

(17) Landsbergen, S. P. J. 1976. Syntax and formal semantics of English in PHLIQA1. In L. Steels (Eds.), *Advances in Natural Language Processing*. Antwerp: University of Antwerp.

(18) Lea, W. (Ed.) 1979. *Trends in Speech Recognition*. Englewood Cliffs, N.J.: Prentice Hall.

(19) Lindsay, R. K. 1963. Inferential memory as the basis of machines which understand natural language. In Feigenbaum and Feldman, pp. 217-233.

(20) Marcus, M. P. 1980. *A Theory of Syntactic Recognition for Natural Language*. Cambridge: MIT Press.

(21) Minsky, M. (Ed.) 1968. *Semantic Information Processing*. Cambridge: MIT Press.

(22) Minsky, M. 1975. A framework for representing knowledge. In P. Winston (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill.

(23) Quillian, M. R. 1968. Semantic memory. In Minsky, pp. 227-270.

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

(25) Raphael, B. 1968. SIR: A computer program for semantic information retrieval. In Minsky, pp. 33-145.

(26) Robinson, A. E., Appelt, D. E., Grosz, B. J., Hendrix, G. G., and Robinson, J. J. 1980. Interpreting natural-language utterances in dialogs about tasks. Technical note #210, Artificial Intelligence Center, SRI International, Menlo Park, CA.

(27) Rustin, R. (Ed.) 1973. *Natural Language Processing*. New York: Algorithmics Press.

(28) Schank, R. 1975. *Conceptual Information Processing*. Amsterdam: North-Holland.

(29) Schank, R., and Abelson, R. P. 1977. *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ : Lawrence Erlbaum.

(30) Schank, R., and Colby, K. (Eds.) 1973. *Computer Models of Thought and Language*. San Francisco: W. H. Freeman.

(31) Schank, R., and Nash-Webber, B. (Eds.) 1975. *Theoretical Issues in Natural Language Processing: An Interdisciplinary Workshop in Computational Linguistics, Psychology, Linguistics, and Artificial Intelligence*. Cambridge, MA.: Bolt Beranek and Newman, Inc.

(32) Simmons, R. F., Klein, S., and McConlogue, K. 1964. Indexing and dependency logic for answering English questions. *American Documentation*, 15, pp. 196-204.

(33) Simmons, R. F., Burger, J. F., and Long, R. E. 1966. An approach toward answering English questions from text. *Proceedings of the AFIPS Fall Joint Computer Conference*, 29. Washington: Spartan Books, pp. 349-356.

(34) Waltz, D. L. (Ed.) 1978. *Theoretical Issues in Natural Language Processing-2*. New York Association for Computing Machinery.

(35) Waltz, D. L. 1977. Natural language interfaces. *SIGART Newsletter*, No. 61, pp. 16-64.

(36) Weaver, W. 1949. Translation. In W. N. Locke and A. D. Booth (Eds.), *Machine Translation of Languages*. New York Technology Press of MIT and John Wiley & Sons, 1955, pp. 15-23.

(37) Weizenbaum, J. 1966. ELIZA—A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9, pp. 36-45.

(38) Weizenbaum, J. 1976. *Computer Power and Human Reason: From Judgement to Calculation*. San Francisco: W. H. Freeman.

(39) Wilks, Y. 1975. A preferential, pattern-seeking semantics for natural language inference. *Artificial Intelligence*, 6, pp. 53-74.

(40) Winograd, T. 1972. *Understanding Natural Language*. New York: Academic Press.

(41) Winogard, T. 1975. Frame representations and the declarative/procedural controversy. In Bobrow and Collins, pp. 185-210.

(42) Winograd, T. Forthcoming. *Language as a Cognitive Process*. Reading, MA: Addison-Wesley.

(43) Woods, W. A. 1973. An experimental parsing system for transition network grammars. In Rustin, pp. 111-154.