

Book Reviews

Associative Networks - Representation and Use of Knowledge by Computers

Nicholas V. Findler, Editor

Academic Press, New York, 1979,
462 pp., \$42.50, ISBN 0-12-256380-8.

Upon opening this book and leafing through the pages, one gets the impression of an important compendium. The fourteen articles provide good coverage of semantic networks and related systems for representing knowledge. Their average length of 33 pages is long enough to give each author reasonable scope, yet short enough to permit a variety of viewpoints to be expressed in a single volume. The editor should be commended for his efforts in putting together a well-organized book instead of just another collection of unrelated papers.

In the first article, *On the Epistemological Status of Semantic Networks*, Ronald Brachman surveys semantic networks since 1966 and introduces many of the issues that are discussed by other authors in later chapters. His main point is the need for greater structure in the nets than a uniform graph of concept nodes and relationship arcs. In his system, KLONE, generic concepts are themselves defined by a network, which is inherited either by individual concepts that are instances of the generic or by other generic concepts that represent subtypes of the higher level generic.

As a survey, Brachman's article is good; but as a history, it is flawed by the common error that semantic networks "originated in the work of Quillian." Yet semantic networks, like much of the AI work on natural language, are a re-invention of what was done for machine translation a decade earlier. In linguistics, the most extensive network representation was the dependency grammar by Lucien Tesnière (1959). Although Tesnière never used a computer himself, he had a strong influence on MT work in Europe. In the United States, dependency parsers were implemented by Hays (1964) and Klein (1965), whom Schank and Tesler (1969) cited as an influence on their work. Of the semantic networks for MT, the most sophisticated were the correlational nets by Silvio Ceccato (1956, 1961, 1962). His work was so advanced that Bar-Hillel (1960) dismissed it as "practically hopeless" because of its obscure speculations about "elements of thought." Contrary to Brachman's claim that Carbonell's nets were the first to distinguish subtypes from instances of a type, Ceccato had distinct relations for *member-class*, *species-genus*, and *part-whole*. Other arcs in Ceccato's nets represented linguistic cases such as *object*, *instrument*, and *result* and spatial relations such as *containing*, *covering*, and *contiguity*. Ceccato implemented the typical AI mechanisms of

inheritance of properties from genus to species and frame-like "constellations" which "indicate both the particular relations that extend from the analyzed thing, and the things that can satisfy these relations" (1961, p. 65). Given the concepts PAINTBRUSH and PICTURE, he could trace a path through a constellation to show that PAINTBRUSH was the instrument of PAINT and PICTURE was the result.

Two other articles in the book introduce additional structure into semantic networks. In *A Procedural Semantics for Semantic Networks*, Hector Levesque and John Mylopoulos develop inheritance mechanisms similar to Brachman's, but with a different notation. Their main innovation is the merger of semantic networks with a network notation for procedures and PLANNER-like conditions for invoking the procedures. Their work forms a bridge between declarative forms of semantic nets and procedural languages for knowledge representation. In *Making Preferences More Active*, Yorick Wilks extends his earlier work on preference semantics with more frame-like structures called *pseudotexts*. He starts with an example from the *London Times* and shows how his system would deal with typical violations of selectional constraints. His analysis is worth reading by anyone who is programming computers to understand natural language, although it is unlikely that others will be quick to adopt Wilks' exuberant terminology of formulas, templates, paraplates, semantic blocks, extractions, and pseudotexts.

Several articles develop notations for representing quantifiers and specifying their scope. In *Encoding Knowledge in Partitioned Networks*, Gary Hendrix presents a general method of nesting subnets inside of other nets. The nesting shows scope of quantifiers, but it could also support other uses for nested contexts such as modalities. Hendrix assumes that the occurrence of a concept automatically asserts its existence; but if a generic concept occurs in both the antecedent and the consequent of an implication, it is assumed to have a universal quantifier. In *Extensional Semantic Networks*, Jürgen Janas and Camilla Schwind also use partitioning to show scope of quantifiers, but they append explicit existential or universal quantifiers to the generic concepts. Despite the title of their article, the authors do represent intensional information in the networks and present a good discussion of the distinction between intensions and extensions. In *The Structure and Organization of a Semantic Net for Comprehension and Inference*, Lenhart Schubert, Randolph Goebel, and Nicholas Cercone introduce various kinds of dotted, dashed, and double links that show scope of Boolean operators, quantifiers, and modality. Their notation is less readable than Hendrix's or Janas and Schwind's, but they discuss general issues of organiz-

ing and retrieving knowledge that would apply to almost any representation. In *The SNePS Semantic Network Processing System*, Stuart Shapiro presents an extensional approach to handling quantifiers with structure-building rules; "Every man loves some woman," for example, corresponds to a rule that builds a structure for "x loves a woman y" whenever it finds a node for a man x. That approach works when all of the facts are represented in a single context. But to represent "Tom believes that every man loves some woman" would require a strange nesting of structure-building rules inside the context of the verb complement of BELIEVE.

In *A Predicate Calculus Based Semantic Network for Deductive Searching*, James McSkimin and Jack Minker present a solid, well-written article on sorted first-order predicate calculus for data base inference. But calling their notation a semantic network is highly misleading. According to Brachman, the common thread that holds together the various forms of semantic networks is their use "in understanding natural language"; and according to Janas and Schwind, "intensionality is undoubtedly a constituent of any semantic network." McSkimin and Minker's approach fails both of these criteria: they make no attempt to deal with natural language, and their semantic categories are "simply labels for a set of objects." With their purely extensional approach, the concept UNICORN would be a subconcept of anything because the set of all unicorns, being empty, is a subset of all other sets.

Three of the articles apply some form of semantic networks to specific problems in computational linguistics. In *Re: The Gettysburg Address*, Roger Schank and Jaime Carbonell, Jr., analyze Lincoln's famous prose as a stimulating challenge for conceptual dependency (CD) theory. As a result of the analysis, they extend the list of primitive CD ACTs with seven social ACTs such as AUTHORIZE, ORDER, and PETITION. The social ACTs are high-level concepts that may optionally be expanded in terms of the low-level primitives. Introducing them solves a common problem of CD theory: too much expansion of concept nodes can lead to a combinatorial explosion of primitive nodes. In *Rule Forms for Verse, Sentences, and Story Trees*, Robert Simmons and Alfred Correia draw some parallels between story generation, problem solving, and theorem proving. As examples, they cite Meehan's Metanovel as well as some of their own work. Although Meehan's underlying representation uses conceptual dependency graphs instead of standard predicate calculus, the output of his story generator may be viewed as a trace of the steps taken by a classical theorem prover. In *On Representing Commonsense Knowledge*, Benjamin Kuipers gives a working definition of commonsense knowledge, discusses some constraints on systems that have it, and then proposes a representation based on a partial order and associative triples. His definition seems acceptable, but the

proposed representation is one of the least structured in the book, and the example he gives is too limited to show that the representation can support the definition.

Three other articles discuss systems that use some form of semantic networks. In *A Heuristic Information Retrieval System Based on Associative Networks*, Nicholas Findler mentions a keyword, ELIZA-like system that he developed, proposes a new information retrieval system called IRUHS-1, describes some clustering techniques for fuzzy retrieval, and then gives a sample dialog that he would like IRUHS-1 to support. Yet the dialog would require sophisticated syntax, semantics, and deductive capabilities that the body of the article never discusses. In *Representations to Aid Distributed Understanding in a Multiprogram System*, Christopher Riesbeck discusses the interactions between the various programs developed by generations of graduate students at the Yale AI Project. The article is not intelligible in isolation, but anyone who has been following the history of MARGIE, SAM, PAM, and QUALM will find an interesting discussion of how one program leaves notes for another one to elaborate. In *Five Aspects of a Full-Scale Story Comprehension Model*, Chuck Rieger proposes an integrated system for understanding a real story, *The Magic Grinder*. Like Schank and Carbonell, he introduces high-level concepts like KISS instead of limiting the system to the primitive ACTs of PTRANSing lips. Since some of his techniques, such as parsing with word sense networks, may have high overhead, his work in testing them on a substantial body of text is necessary to determine their practicality.

Various forms of { associative | conceptual | cognitive | partitioned | semantic | structured } [dependency | inheritance] {graphs | nets | networks} have major differences in notation, terminology, and expressive power, but they develop similar network solutions for their common problems of representing knowledge in processing natural language. Yet many areas remain to be developed:

- None of the systems have a good analog of the LISP QUOTE operator. Such a feature would enable a system to distinguish between believing a proposition and actually changing a model of the world. A QUOTE operator is also necessary for handling the phrase "dedicated to the proposition that..." in Schank and Carbonell's example.
- Many of the systems have special case inference rules for plausible or exact deduction, but questions of completeness and consistency are usually ignored. Their relationship to various nonstandard forms of logic (Leblanc 1973) must also be investigated.
- Procedural techniques like Levesque and Mylopoulos's will have to be refined and tested on more complex examples than factorial.

- Benign neglect of syntax was once a healthy reaction to transformational grammar, but the field is now mature enough to accommodate more syntactic sophistication.
- And for practical systems, naturalness, readability, and ease of use become crucial.

In summary, this book is an exciting record of work in progress. It assumes too much background material to be used as an introductory textbook, but it could be used in a seminar course on knowledge representation, either with supplementary lectures on logic and philosophy or with programming exercises that show students how to implement such networks. By presenting fourteen systems in one volume, it invites detailed comparisons that can help knowledge engineers select the best features for their future designs.

John F. Sowa, IBM Systems Research Institute

References

- Bar-Hillel, Yehoshua (1960) "The present status of automatic translation of languages," in F. L. Alt, ed., *Advances in Computers*, vol. 1, Academic Press, New York, pp. 91-163.
- Ceccato, Silvio (1956) "La grammatica insegnata alle machine," *Civiltà delle Machine*, Nos. 1 & 2.
- Ceccato, Silvio (1961) *Linguistic Analysis and Programming for Mechanical Translation*, Gordon and Breach, New York.
- Ceccato, Silvio (1962) "Automatic translation of languages," presented at the NATO Summer School, Venice, July 1962; reprinted in *Information Storage and Retrieval*, vol. 2, no. 3, pp. 105-158, 1964.
- Hays, David G. (1964) "Dependency theory: a formalism and some observations," *Language*, vol. 40, no. 4, pp. 511-525.
- Klein, Sheldon (1965) "Automatic paraphrasing in essay format," *Mechanical Translation*, vol. 8, pp. 68-83.
- Leblanc, Hughes, ed. (1973) *Truth, Syntax, and Modality*, North-Holland Publishing Co., Amsterdam.
- Schank, Roger C., & Lawrence G. Tesler (1969) "A conceptual parser for natural language," *Proceedings of the First IJCAI*, pp. 569-578.
- Tesnière, Lucien (1959) *Éléments de Syntaxe Structurale*, Librairie C. Klincksieck, Paris, second edition 1965.

Discourse Production - A Computer Model of Some Aspects of a Speaker

Anthony Davey

Edinburgh Univ. Press, Scotland, 1978,
170 pp., \$16.00, ISBN 0-85224-339-1.

"This book describes a computer program that produces English discourse. The program is capable of describing in a sequence of English sentences any game of noughts and crosses (tic-tac-toe), whether given or actually played with the program." (From the Preface.)

The game descriptions have several properties that make them non-trivial. The entities of the game are referred to by fairly natural English noun phrases, making use of anaphora and standard constructions for qualification (such as relative clauses), and taking advantage of symmetry of the board. In many cases, moves are described for their strategic value, and some mistakes of the program's opponent are mentioned. The following sentence produced by the program illustrates these properties. *If you had blocked my line, you would have threatened me, but you took the corner adjacent to the one which you took first and so I won by completing my line.* This also illustrates that the program uses connectors like *and*, *but*, *so*, and *although*, to string clauses together in coherent discourse.

The grammatical theory used by the author is systemic grammar, essentially the version developed by R. A. Hudson in *English Complex Sentences*, North-Holland, 1971. In this theory, a grammatical item (such as a clause) is classified by an associated bundle

of features. *Feature-realization* rules determine a set of *functions* (like SUBJECT) for constituents. *Structure-building* rules manipulate functions, gathering them into bundles, one for each constituent. *Function-realization* rules associate feature bundles with these function bundles, and the cycle continues, down to the word level. On each level there is some freedom of choice for the features of an item, but there is a network of constraints on co-occurring features called the *system-network*.

In Davey's adaptation, semantic representations are carried along with the above sort of syntactic representations, and "specialist" procedures are used to determine features of grammatical items where there is freedom left in the system-networks. The overall control, however, is not strictly top-down as in Hudson's system described above. Sometimes specialist procedures actually construct the English text for an item *before* determining all of its features.

It is unfortunate that the book did not reach published form earlier. The work was done in the period 1970-1973 as the author's thesis at Edinburgh University. A chapter devoted to *A Review of Previous Systems* essentially covers only what was published by 1972. The program does not take account of advances made in systemic grammar since around 1972. Some of the constructions made by the program seem less direct than they should be, and this would be improved in the light of later developments.

Michael C. McCord, University of Kentucky