6

Semantic Representations

THE COMPUTER PROGRAMS I HAVE DESCRIBED SO FAR PERFORMED TRANSFORMA-tions on relatively simple symbol structures, which were all that were required for the mathematical problems, puzzles, and games that these programs dealt with. The main effort was in coming up with and using problem-specific heuristics (such as features to be used in computing the value of a checkers position, for example) to limit the number of transformations of these structures in searches for solutions. As Minsky put it, "The most central idea of the pre-1962 period was that of finding heuristic devices to control the breadth of a *trial-and-error search*."[1] In the early 1960s, several Ph.D. research projects, some performed under Minsky's direction at MIT, began to employ more complex symbol structures in programs for performing various intellectual tasks. Because of their rich, articulated content of information about their problem topic, these structures were called *semantic representations*.[2] As Minsky wrote, "Within the small domain in which each program operates, the performance [of these programs] is not too bad compared with some human activities. . . . But much more important than what these particular experiments achieve are the methods they use to achieve what they do, *for each is a first trial of previously untested ideas*."[3] I'll describe some examples of these sorts of projects and the new methods that they employed.

6.1 Solving Geometric Analogy Problems

Thomas G. Evans (1934–) programmed a system that was able to perform well on some standard geometric analogy tests. It was apparently the largest program written up to that time in John McCarthy's new programming language, LISP (which I'll describe later). I quote from an article based on Evans's 1963 dissertation, which presented this work:[4]

> We shall be considering the solution by machine of so-called "geometric-analogy" intelligence-test questions. Each member of this class of problems consists of a set of labeled line drawings. The task to be performed can be described by the question: "Figure A is to Figure B as Figure C is to which of the following figures?" For example [in Fig. 6.1] it seems safe to say that most people would agree with the program we are about to describe, in choosing [number 4] as the desired answer.

He further noted that "problems of this type are widely regarded as requiring a high degree of intelligence for their solution and in fact are used as a touchstone of intelligence in some general intelligence tests used for college admission and other

https://doi.org/10.1017/CBO9780511819346.008 Published online by Cambridge University Press

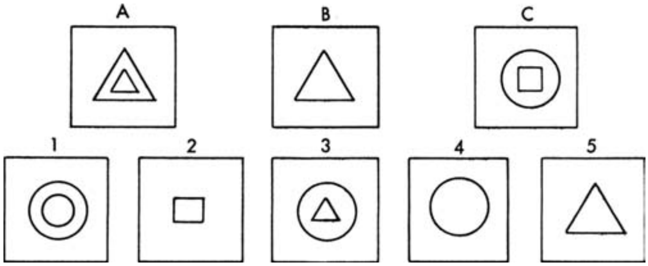
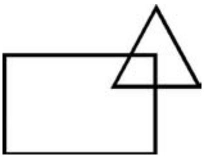


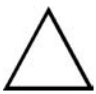
Figure 6.1. An analogy problem.

purposes." So, again, AI research concentrated on mechanizing tasks requiring human intelligence.

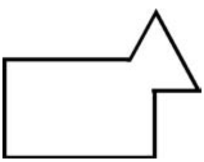
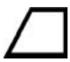
Evans's program first transformed the diagrams presented to it so that they revealed how they were composed out of parts. He called these "articular" representations. Of the possibly several decompositions possible, the one chosen by the program depended on its "context." (This choice is one example of a heuristic used by the program.) For example, the diagram



could either be decomposed into

 and 

or into

 and 

But if the analogy problem contained another diagram (part of the context):

then the first decomposition would be chosen.

https://doi.org/10.1017/CBO9780511819346.008 Published online by Cambridge University Press

Evans represented diagrams and their parts as complex symbol structures consisting of rather elaborate combinations of lists and lists of lists whose elements indicated which parts were inside or outside (or above or below) which other parts, and so on. Those details need not concern us here, but they did allow Evans to specify "rules" for his program that could be used to show how one diagram could be transformed into another. The program was able to infer which combinations of these rules transformed Figure A of a given problem into Figure B. Then it could apply this transformation to Figure C. If one of the multiple-choice answers resulted, it would give that one as its answer. Otherwise, the program "weakened" the transformation just enough so that one of the answers was produced, and that would be the program's answer.

Evans summarized his results as follows:

> Allowing ourselves only [the parts of the program actually implemented], our estimate would be that of the 30 geometric-analogy problems on a typical edition of the ACE tests, [the program] can successfully solve at least 15 and possibly as many as 20 problems.

He notes that this level of performance compares favorably with the average high school student.

6.2 Storing Information and Answering Questions

Another of Minsky's Ph.D. students during the early 1960s, Bertram Raphael (1936–), focused on the problem of "machine understanding." In his dissertation,[5] Raphael explained that

> a computer should be considered able to "understand" if it can converse intelligently, i.e., if it can remember what it is told, answer questions, and make responses which a human observer considers reasonable.

Raphael wanted to be able to tell things to a computer and then ask it questions whose answers could be deduced from the things it had been told. (The telling and asking were to be accomplished by typing sentences and queries.) Here are some examples of the kinds of things he wanted to tell it:

Every boy is a person.
A finger is part of a hand.
There are two hands on each person.
John is a boy.
Every hand has five fingers.

Given this information, Raphael would want his system to be able to deduce the answer to the question "How many fingers does John have?"

Because Raphael wanted his system to communicate with people, he wanted its input and output languages to be "reasonably close to natural English." He recognized that "the linguistic problem of transforming natural language input into a usable form will have to be solved before we obtain a general semantic information retrieval system." This "linguistic problem" is quite difficult and still not "solved" even though much progress has been made since the 1960s. Raphael used various

https://doi.org/10.1017/CBO9780511819346.008 Published online by Cambridge University Press

"devices" (as he called them and which are not germane to our present discussion) to "bypass [the general problem of dealing with natural language] while still utilizing understandable English-like input and output."

The main problem that Raphael attacked was how to organize facts in the computer's memory so that the relevant deductions could be made. As Raphael put it, "The most important prerequisite for the ability to 'understand' is a suitable internal representation, or model, for stored information. The model should be structured so that information relevant for question-answering is easily accessible."[6]

Raphael called his system SIR, for Semantic Information Retrieval, (which he programmed in LISP). He used the word "semantic" because SIR modeled sentences in a way dependent on their meanings. The sentences that SIR could deal with involved "entities" (such as John, boy, hand, finger, and so on) and relations among these entities (such as "set-membership," "part–whole," "ownership," "above," "beside," and other spatial relationships). The model, then, had to have ways for representing entities and the relationships among them.

Entities such as John and boy were represented by the LISP computer words `JOHN` and `BOY`, respectively. (Of course, the computer had no way of knowing that the computer word `JOHN` had anything to do with the person John. Raphael could have just as well represented John in the computer by `X13F27` so long as he used that representation consistently for John. Using the computer word `JOHN` was a mnemonic convenience for the programmer – not for the computer!) When representing the fact that John is a boy, SIR would "link" a computer expression `(SUPER-SET JOHN BOY)` to the expression `JOHN` and link a computer expression `(SUB-SET BOY JOHN)` to the expression `BOY`. Thus, if SIR were asked to name a boy, it could reply "`JOHN`" by referring to `BOY` in its model, looking at its `SUB-SET` link and retrieving `JOHN`. (I have simplified the representations somewhat to get the main ideas across; SIR's actual representations were a bit more complicated.)

SIR could deal with dozens of different entities and relations among them. Every time it was told new information, it would add new entities and links as needed. It also had several mechanisms for making logical deductions and for doing simple arithmetic. The very structure of the model facilitated many of its deductions because, as Minsky pointed out in his discussion of Raphael's thesis, "the direct predicate-links . . . almost physically chain together the immediate logical consequences of the given information."[7]

SIR was also the first AI system to use the "exception principle" in reasoning. This principle is best explained by quoting directly from Raphael's thesis:

> General information about "all the elements" of a set is considered to apply to particular elements only in the absence of more specific information about those elements. Thus it is not necessarily contradictory to learn that "mammals are land animals" and yet "a whale is a mammal which always lives in water." In the program, this idea is implemented by always referring for desired information to the property-list [that is, links] of the individual concerned *before* looking at the descriptions of sets to which the individual belongs.
>
> The justification for this departure from the no-exception principles of Aristotelian logic is that this precedence of specific facts over background knowledge seems to be the way people operate, and I wish the computer to communicate with people as naturally as possible.

The present program does not experience the uncomfortable feeling people frequently get when they must face facts like "a whale is a mammal which lives in water although mammals as a rule live on land."

The exception principle was studied by AI researchers in much more detail later and led to what is called default reasoning and nonmonotonic logics, as we shall see.

6.3 Semantic Networks

It is instructive to think of SIR's representational scheme in terms of a network. The entities (such as JOHN and BOY) are the "nodes" of the network, and the relational links (such as SUB-SET) are the connections between nodes. SIR was an early version of what would become an important representational idea in artificial intelligence, namely, *semantic networks*. It was not the first, however. John Sowa, who has written extensively about semantic networks, claims that the "oldest known semantic network was drawn in the 3rd century AD by the Greek philosopher Porphyry in his commentary on Aristotle's categories."[8] In 1961 Margaret Masterman (1910–1986), Director of the Cambridge Language Research Unit, used a semantic network in a translation system in which concepts were ordered in a hierarchy.[9]

M. Ross Quillian, a student of Herb Simon's at the Carnegie Institute of Technology, was interested, along with Newell and Simon, in computational models of human mental processes, specifically memory organization. He developed a memory model consisting of a semantic network of nodes representing English words. The nodes were interconnected by what he called "associative links." In Quillian's words, "In the memory model, ingredients used to build up a concept are represented by the token nodes naming other concepts, while the configurational meaning of the concept is represented by the particular structure of interlinkages connecting those token nodes to each other."

Quillian goes on to write that "[t]he central question asked in this research has been: What constitutes a reasonable view of how semantic information is organized within a person's memory? In other words: What sort of representational format can permit the 'meanings' of words to be stored, so that humanlike use of these meanings is possible?"[10]

I can illustrate how Quillian's network format represents meaning by using one of his examples. Consider the different meanings of the word "plant." One such meaning is given by linking the node PLANT to other nodes, such as LIVE, LEAF, FOOD, AIR, WATER, and EARTH, through connections that represent that a plant (according to this meaning of the word) is alive, has leaves, and gets its food from air, water, and earth. Another meaning of "plant" links PLANT to other nodes, such as PEOPLE, PROCESS, and INDUSTRY, through connections that represent that a plant (according to this other meaning of the word) is an apparatus that uses people for engaging in a process used in industry.

According to Quillian, the meaning of a term is represented by its place in the network and how it is connected to other terms. This same idea is used in dictionaries where the meaning of a word is given by mentioning the relationship of this word to other words. The meanings of those other words are, in turn, given by their relationships to yet other words. So we can think of a dictionary as being like a large semantic network of words linked to other words.

By using this view, the *full* meaning of a concept can be quite extensive. As Quillian puts it,

> Suppose that a person were asked to state everything he knows about the concept "machine." . . . This information will start off with the more "compelling" facts about machines, such as that they are usually man-made, involve moving parts, and so on, and will proceed "down" to less and less inclusive facts, such as the fact that typewriters are machines, and then eventually will get to much more remote information about machines, such as the fact that a typewriter has a stop which prevents its carriage from flying off each time it is returned. We are suggesting that this information can all usefully be viewed as part of the subject's concept of "machine."

In what way is Quillian's network a model of human memory organization? Quillian explored two capabilities of human memory modeled by his network. One was comparing and contrasting two different words. Quillian proposed that this be done by a process that came to be called "spreading activation." Conceptually, one starts at the nodes representing the two words and gradually traverses the links emanating from them, "activating" the nodes along the way. This process continues until the two "waves" of activation intersect, thus producing a "path" between the two original nodes. Quillian proposed that the total "distance" along this path between the two words could be used as a measure of their similarity. The path can be used to produce an account comparing the two words. (Quillian's program had mechanisms for expressing this account in simple English.)

To use one of Quillian's examples, suppose we wanted to compare the words "cry" and "comfort." The spreading activations would intersect at the word "sad," and the English account would express something like "to cry is to make a sad sound, and to comfort is to make something less sad."

Quillian was also interested in how the network could be used to "disambiguate" two possible uses of the same word. Consider, for example, the sentence "After the strike, the president sent him away." The network can encode different meanings of the word "strike." One such might involve a labor dispute, another might involve baseball, and yet another involve a raid by military aircraft. Which of these meanings is intended by the sentence? Presumably, activation proceeding outward from the word "president" would eventually reach concepts having to do with labor disputes before reaching concepts having to do with baseball or the military. Thus, the "labor dispute" meaning would be preferred because it is "closer," given that the word "president" is in the sentence. In contrast, a different conclusion would be reached for the sentence "After the strike, the umpire sent him away."

Quillian's model differs from some later semantic networks in that it does not have a predetermined hierarchy of superclasses and subclasses. As Quillian puts it, "every word is the patriarch of its own separate hierarchy *when some search process starts with it*. Similarly, every word lies at various places down within the hierarchies of (i.e., is an ingredient in) a great many other word concepts, when processing starts with them."

Notes

1. Marvin Minsky (ed.), "Introduction," *Semantic Information Processing*, p. 9, Cambridge, MA: MIT Press, 1968. [96]
2. It might be argued that the diagram used by Gelernter's Gelernter, Herb geometry program was an earlier use of a semantic representation. [96]
3. Marvin Minsky, *op. cit.*, p. 1. [96]
4. Thomas G. Evans, "A Program for the Solution of a Class of Geometric-Analogy Intelligence-Test Questions," in Marvin L. Minsky, *op. cit.*, p. 271. [96]
5. Bertram Raphael, "SIR: Semantic Information Retrieval," in Marvin Minsky, *op. cit.*, pp. 33–145. (This is a partial reprint of his 1964 Ph.D. dissertation.) [98]
6. Marvin Minsky, *op. cit.*, p. 35. [99]
7. Marvin Minsky, *op. cit.*, p. 17. [99]
8. From an article by John F. Sowa at http://www.jfsowa.com/pubs/semnet.htm. (This is a revised and extended version of one that was originally written for the *Encyclopedia of Artificial Intelligence*, edited by Stuart C. Shapiro, Wiley, 1987, second edition, 1992.) [100]
9. Margaret Masterman, "Semantic Message Detection for Machine Translation, Using an Interlingua," in *Proceedings of the 1961 International Conference on Machine Translation of Languages and Applied Language Analysis*, pp. 438–475, London: Her Majesty's Stationery Office, 1962. [100]
10. M. Ross Quillian, "Semantic Memory," Ph.D. dissertation, Carnegie Institute of Technology (now Carnegie Mellon University), October 1966. (This work also appears as Report AFCRL-66-189 and is partially reprinted in M. Minsky (ed.), *Semantic Information Processing*, pp. 216–270, Cambridge, MA: MIT Press, 1968.) [100]