Reprinted from the IEEE TRANSACTIONS OF THE
PROFESSIONAL TECHNICAL GROUP ON *INFORMATION THEORY*
Volume IT-9, Number 4, October 1963
PRINTED IN THE U.S.A.

ARTIFICIAL INTELLIGENCE RESEARCH

EDWARD A. FEIGENBAUM*

Though we humans are facile at applying the adjective *intelligent* to the behavior of other human beings, we do so with great subjectivity. Attempts to obtain precise measures of intelligence in the absence of a concrete model of intellective processes have ended as exercises in scaling and correlation of test scores with other "real world" performance data. We must be content, at present, to pursue the definition of the goals of artificial intelligence research at the level of human subjective judgment on which we ordinarily operate in these matters. This is, in fact, the essence of Turing's proposal for "testing" the thinking capabilities of a machine [1].

The following loose statement of goals has been advanced, in one form or another, by a number of researchers in the field [2], [3]: artificial intelligence research is concerned with constructing machines (usually programs for general-purpose computers) which exhibit behavior such that, if it were observed in human activity, we would deign to label the behavior "intelligent." Researchers in the field hold to the working hypothesis that human thinking is wholly information-processing activity within the human nervous system; that ultimately

* University of California, Berkeley, Calif.

these information processes are perfectly explicable; that the road to this explication lies in observation, experimentation, analysis, modeling, model validation, etc.; and that digital computers, being general information-processing devices, can be programmed to carry out any and all of the information processes thus explicated. Researchers differ considerably over the time scale envisioned for such a program [4].

Two subgoals in the research are discernable, though the fine line which separates them is, in fact, blurred. One group of researchers is concerned with simulating human information-processing activity, with the quest for precise psychological theories of human cognitive activity. These researchers use the digital computer as their modeling medium. The computer is the tool for exercising the information-processing model, for generating the remote and complex consequences of the mechanisms postulated in the model. Intelligent behavior of the model is not valued for its own sake; it has value only insofar as it correctly predicts human behavior in the same tasks. A second group of researchers is concerned with evoking intelligent behavior from machines *whether or not* the information processes employed have anything to do with plausible human cognitive mechanisms. Here the elicited behavior is, in a sense, an end in itself. These researchers are free to employ in their designs the most powerful mathematical, logical, and computer-oriented techniques available. Needless to say, there has been a great deal of cross insemination of ideas and borrowing of techniques between the two groups.

Minsky [5] has produced a review of the early literature in the artificial intelligence area; it is required reading for any serious student of this research. This short report will not attempt to duplicate Minsky's work, but will be in the nature of an epilogue to Minsky's review. Minsky himself has produced one such epilogue [6]. However, a few of the topics covered by Minsky are reviewed again here.

Historical Perspective

A brief historical resume, highlighting a few important research events before 1960, may be useful. After much discussion in the early 1950's about the potential of computers for intelligent behavior, two concrete computer programs emerged which set the stage for subsequent development: Selfridge's and Dineen's pattern-recognition program [7] and the Logic Theory program of Newell, Shaw, and Simon [8]. The latter, a program which proved theorems from the Whitehead and Russell "Principia Mathematica," introduced the idea of heuristic problem-solving methods as a means of organizing and administering the complex problem-solving activity by machines [9]. Gelernter and his associates subsequently expanded on this theme in their Geometry-Theorem-Proving Program [10], especially in the use of the diagram as a heuristic device.

In this early period, a number of game-playing programs were written—notably a checker-playing program by Samuel [11], and chess-playing programs by a Los Alamos group, by Bernstein, and by Newell, Shaw, and Simon (see [12] for a review and appraisal of these programs). Samuel made a serious attempt at embedding processes for machine learning in his checker program. He tried two types of learning: a form of "rote" learning of positions; and a form of "generalization" learning involving adjustment of parameters in an evaluation function.

Finally in this period, the initial formulation of the General Problem-Solving Program of Newell, Shaw, and Simon [13] was accomplished.

General Problem-Solver

The General Problem-Solving Program (GPS) overlaps the border between research on machine problem-solving and simulation of human cognitive processes. On the one hand, it represents an attempt to state precisely an information-processing theory of human problem-solving processes [14]. On the other hand, it represents a search for fundamental mechanisms that will endow computers with the "general-purpose" problem-solving character that we associate with intelligence.

GPS consists of a core of heuristic problem-solving processes that are not task-specific, surrounded by information processes and structures for description and manipulation of some specific "task environment." Given an adequate description of a task (*e.g.*, manipulating logic expressions, proving trigonometric identities) in a particular form, the core processes of GPS will solve (or at least attempt to solve) problems of that kind. It is in this sense that GPS is "general."

GPS solves problems which can be cast in the form: given an initial *object* (or state) A, a target object (or state) B, and a set of *operators* (or rules) which transform objects into other objects, *discover a sequence of operators which will transform A into B*. The principal method employed is the so-called means-ends analysis or, as it has sometimes been called, "functional reasoning." Another method, called the "planning" method, has also undergone experimentation. This method involves the use of the GPS means-ends processes to discover "plans" (possible operator sequences) for problem solution in an abstracted problem space of highly simplified objects and operators [13].

Attempts to achieve prediction-in-depth of human "thinking aloud" protocols tape-recorded during problem-solving have been rather successful [15]. The GPS means-ends method has also been used as the basis for other problem-solving programs, notably Simon's Heuristic Compiler [16], a program which writes programs from state-language or process-language descriptions of the required information processes.

GPS is basically a model of performance processes in problem-solving. However, mechanisms have been explored, and partially implemented, which enable GPS to learn: 1) the associations between operators and the functions they perform in reducing differences between

objects; and 2) those differences between objects which are relevant to effective problem-solving in a given task [17], [18].

Game-Playing Programs

Appreciation of research on game-playing programs as steps toward artificial intelligence has increased [3] at the same time that the research itself has waned. Most chess-playing programs are either moribund or prenatal. Samuel's Checker-Playing Program [11] continues to play (and win) games and undergoes modest change. It achieved a notable milestone in 1962 when it defeated a well-known checker player [19].

McCarthy and his students have developed a chess-playing program with a particularly simple search strategy. It builds an analysis tree N (max.) half-moves deep (*e.g.*, 8 half-moves deep, max.) by the following method: at a node at level i, examine and evaluate all possible immediate moves, selecting k_i "best" moves for further exploration. In one particular realization, the k_i were set as 4–3–2–1–1–1–1–1. In 1962 this chess program decisively defeated a good chess player who spotted the program a Queen [20].

Question-Answering Machines

This research is concerned with attempts to model processes of "comprehension" (as we understand this term with its full human connotations) of natural language questions and text. The programs mentioned below differ in the scope of their universe of discourse and in the sophistication of the processes employed in answering questions about this universe. The SAD SAM program of Lindsay [21] deals only with kinship relationships. Its input language is a simplified form of English called Basic English. It reads sentences about family relationships, and from these sentences it constructs an internal model of the family under discussion. When questioned, it uses this model to give answers which were implicit (but *not* explicitly stated) in the input sentences.

The data universe for the BASEBALL program of Green, Wolf, Chomsky, and Laughery [22] is a handbook of baseball statistics, organized as a complex list structure. The input questions are in natural English, with some constraints on the type of question that can be asked. In the analysis phase of answering a question, the program constructs a "spec list" for the answer; in the search phase, the "question marks" in the spec list are eliminated in successive stages. In one particularly interesting demonstration, the program was asked five different questions which people easily understand to be equivalent queries, though the questions themselves were quite variously phrased. The program produced equivalent spec lists for all the questions, and therefore the same answer to all.

Raphael [23] has written a program which accepts statements about relationships among objects and produces answers to questions about other relationships among objects, even if the obtaining of the answer requires a complex deduction. Consider this simple example. Suppose, among many statements, the following have been made: John is a man; a man has two arms; an arm has one hand; a hand has five fingers. If the question is asked, "How many fingers has John?", the program will answer "Ten." When the answer is ambiguous, the program will attempt to elicit further statements by appropriately questioning its user.

The SYNTHEX program of Simmons and Klein [24] deals with the broadest data universe, the Child's Golden Book Encyclopedia. The input questions are in natural English. The answers (in the early SYNTHEX models) are text quotations which bear (hopefully) on the answer to the question. The program searches the text using a kind of "answer image," built primarily out of significant words in the question. Later models have incorporated programs of considerable linguistic sophistication, including a program for generating coherent discourse [25].

Other Problem-Solvers and Applications

A number of problem-solving programs are available as illustrations of the power of the heuristic programming techniques that have been developed in the last few years.

Slagle has written a program, called SAINT [26], that performs symbolic analytic integration over a wide range of possible integrands (some very complicated, and difficult for humans to integrate). SAINT's basic problem-solving organization closely resembles that of the Logic Theory Machine. SAINT's procedure involves selecting and applying various simplifications, transformations, and "tricks" to reduce the integrand to a simple form.

In the area of management science research, three heuristic problem-solving programs deserve mention. Tonge has constructed a heuristic program for assembly line balancing [27], a complex combinatorial problem which has proven, in practical applications, to be intractable of solution by ordinary mathematical programming techniques. The problem involves the assignment of men and jobs at work stations, subject to a network of assembly constraints, in such a way as to maximize the rate of assembly. Kuehn and Hamburger [28] have written a heuristic program to solve the problem of locating warehouses in places that will minimize costs in a distribution network, a problem which has not yielded readily to standard mathematical programming techniques. Finally, Clarkson has carried out a study of the decision-making processes of a trust investment officer in a bank, and has developed a heuristic program for selection of common stock portfolios [29]. Clarkson's program is discussed later.

Evans [30] has written a heuristic program to solve geometric analogy problems of a type commonly found on intelligence tests. The problems are of the form, "A is to B as C is to: D, E, F, G, or H?", where the A, B, etc., are actually abstract geometric diagrams, (*e.g.*, point inside circle at lower left corner of diagram). The program uses a language appropriate to the description of simple, two-dimensional geometric figures; forms in this language a set of hypotheses about the relationship between A and B; and applies these hypotheses to C and the candidate list in an attempt to find a single candidate which satisfies the hypothesized A–B relations.

Simulation of Human Cognitive Processes

The goals of this area of research on intelligent machines were described earlier. Research efforts range widely, from a model of problem-solving behavior (the General Problem Solver) to a model of the information-processing behavior of neurotic individuals [31]. As it multiplies from year to year, the research is having a significant growing impact on modern psychological theory. Four of the better-known efforts are described briefly below.

The Elementary Perceiver and Memorizer (EPAM) of Feigenbaum and Simon [32] is a model of the information processes and structures which underlie acquisition, discrimination, and association of elementary symbolic material by humans. The primary information structure of the EPAM associative memory is the discrimination net. This is a type of decoding tree grown by discrimination learning processes during the course of learning to provide uniquely discriminated terminals for storing encoded images of new stimuli. Associations between stimuli are learned by storing response cues with stimulus images; associations are performed by sorting these cues through the discrimination net to select the cued response.

The EPAM program has been run as an "artificial subject" in numerous experiments to elicit and study its behavior as compared with human behavior in the same experiments. Minsky points out [6] that the discrimination learning processes of EPAM have implications for the construction of certain adaptive pattern recognition models.

EPAM-like discrimination trees (that grow in response to changing stimuli) have also been used by Hunt [33] in his model of the processes of human concept attainment. In the concept attainment task, the subject is shown a sequence of stimuli which he is told are, or are not, examples of the concept the experimenter has in mind. The "concept" that the model attains is, in fact, a tree which will correctly sort new stimuli into the correct concept class.

Feldman has constructed a model of how humans develop hypotheses about events in a binary event series, and how they use and modify these hypotheses in predicting succeeding binary events [34]. The hypotheses constructed represent direct "explanations" by the program of how its environment is patterned. Variants of the basic model, tailored to model the behavior of particular human subjects, have been run in numerous simulated binary choice experiments. These models do an excellent job of predicting not only the subjects' actual guesses, but their reasons for making the particular guesses (*i.e.*, their hypotheses about the patterning of the event series).

Clarkson has developed a model of the decision-making behavior of a trust investment officer in a local Pittsburgh bank [29]. Initially, Clarkson studied in great detail the information-processing activities of this individual—his information-gathering and encoding behavior, his noticing and problem-solving processes, his decision making "rules of thumb," etc. He then modeled these processes in a computer program for selecting common stock portfolios, a major function in this bank officer's job. The program was used to predict the officer's portfolio decisions in a subsequent calendar quarter. In the four cases tested, the agreement between the model's portfolio decisions and those of the bank officer was almost complete.

Some Inductive Systems

In the context of artificial intelligence research, an inductive machine is one which, when given a set of evidence, will find a hypothesis that accounts for the evidence (alternatively, will construct a model adequate to explain the observed phenomenon). The hypothesis, or the model, can be used to predict future states of the environment. The classical limitation applies: the machine can never *prove* that its hypothesis is correct, but can only gain increasing confidence in its validity.

Lindsay's SAD SAM program, already discussed, is an example of a program that builds and uses a model of its environment (in this case, the family being discussed in the input sentences). Lindsay has pointed out [21] that a good model is one which is "inferentially rich": implicit in the structure of the model is a great deal of information about the environment (how it is organized, how it behaves, etc.) that was not explicit in the input that gave rise to the model.

Feldman's program, mentioned previously, and Kochen's programs [35], are also systems in this class—they are capable of generating hypotheses about how a simple environment is structured.

Some programs do not build these structural models. For example, Fogel's program builds complex Markovian models [36].

Simon and Kotovsky [37] have written a program which extrapolates letter sequences (the kind of problem found on many intelligence tests). The program searches for structure in the input sequence, encodes what structure it finds into a symbolic model, and uses the model to extrapolate the sequence.

The heuristic program which guides the behavior of Ernst's mechanical hand, MH-1, [38] is capable of acquiring and using models of the objects in its environment. Ernst refers to these models as "abstract images" of the real world being manipulated by the hand. Models of the objects can be given to the program, or the program can induce the models by scanning and "measuring" the environment with the hand. The important point is that the total model of the environment is inferentially rich. Ernst remarks, "It is evident that in some cases the hand really does the reasonable thing if unexpected disturbances of a considerable logical complexity arise, without having to be told so explicitly by the programmer," ([38], p. 34).

Finally, Amarel [39] has exhibited an ingenious program which constructs a "theory" of a process, or transformation. The transformation is made known to the machine by example in the black-box sense, *i.e.*, the machine is shown a set of inputs to the transformation, and the set of outputs which the transformation produces for these inputs. The job of Amarel's machine is to write a program which simulates the behavior of the transformation in the sense that the input-output behavior of this program will be the same as the input-output behavior seen in the examples. Of course, the number of examples shown the machine will not, in general, exhaust the transformation's possibilities, so that the machine is usually working with limited evidence of the behavior of the transformation. At various points in time, additional examples can be added to the example set (augmenting the evidence). The machine attempts to cope with the expanded set of evidence by elaborating its program ("theory") if the program it has does not "work" for the new input-output examples.

Epilogue

This survey has focused primarily on that artificial intelligence research which loosely can be styled "complex cognitive models" to contrast with research on neural-net models, perceptron-type models, simple learning mechanisms, pattern recognition models, etc. Much of the research not treated here is surveyed in accompanying articles.

This reviewer makes no apologies for the sharpness of his focus on cognitive models. This approach has far outdistanced all others to date in progress toward the goal of producing mechanically the kind of behavior labeled "intelligent" in humans. No one yet has demonstrated, or even seriously proposed, a self-organizing neural net which, for example, will play championship checkers or simulate the decision-making behavior of a trust investment officer.

Bibliographies

A number of bibliographies of artificial intelligence research are available. Minsky's descriptor-indexed bibliography [40] and its revision [41] are excellent and recommended. Other comprehensive bibliographies of considerable value are those prepared by the Jet Propulsion Laboratory [42], and by the American Institute of Research [43].

Bibliography

[1] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, New Series 236, pp. 433–460; 1950.
[2] E. Feigenbaum and J. Feldman, Eds., "Computers and Thought," McGraw-Hill Book Company, Inc., New York, N. Y.; 1963.
[3] J. C. R. Licklider, "Interactions Between Artificial Intelligence, Military Intelligence and Command and Control," in preprints of the 1st Congress of the Information System Sciences, The MITRE Corp., Bedford, Mass.; 1962.
[4] J. L. Kelly, Jr., and O. G. Selfridge, "Sophistication in computers: a disagreement," IRE Trans. on Information Theory, vol. IT-8, pp. 78–80; February, 1962.
[5] M. Minsky, "Steps toward artificial intelligence," Proc. IRE, vol. 49, pp. 8–30; January, 1961.
[6] M. Minsky, "Appendix to Steps Toward Artificial Intelligence," in preprints of the 1st Congress of the Information System Sciences, The MITRE Corp., Bedford, Mass., 1962.
[7] O. Selfridge, "Pattern recognition and modern computers,' *Proc. 1955 Western Joint Computer Conf.*, pp. 91–93.
[8] A Newell and H. A. Simon, "The logic theory machine—a complex information processing system," IRE Trans. on Information Theory, vol. IT-2, pp. 61–79; September, 1956
[9] A. Newell, J. C. Shaw, and H. A. Simon, "Empirical explorations of the logic theory machine: a case study in heuristics," *Proc. 1957 Western Joint Computer Conf.*, pp. 218–230.
[10] H. Gelernter, "Realization of a geometry theorem-proving machine," *Proc. Internat'l Conf. on Information Processing*, UNESCO House, Paris, France; 1959.
[11] A. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. & Dev.*, vol. 3, No. 3, pp. 210–229; 1959.
[12] A. Newell, J. C. Shaw, and H. A. Simon, "Chess-playing programs and the problem of complexity," *IBM J. Res. & Dev.*, vol. 2, No. 4, pp. 320–335; 1958.
[13] A. Newell, J. C. Shaw, and H. A. Simon, "Report on a general problem-solving program," *Proc. Internat'l Conf. on Information Processing*, UNESCO House, Paris, France, pp. 256–264; 1959.
[14] A. Newell and H. A. Simon, "Computer simulation of human thought," *Science*, vol. 134, p. 2011; December 22, 1961.
[15] A. Newell and H. A. Simon, "A Program that Simulates Human Thought," in "Lernende Automaten," H. Billing, Ed., Oldenbourg, Munich, Germany; 1961.
[16] H. A. Simon, "The Heuristic Compiler," The RAND Corp., Santa Monica, Calif., RM-3588PR.
[17] A. Newell and H. A. Simon, "A Variety of Intelligent Learning in a General Problem Solver," in "Self-Organizing Systems," M. Yovitts and S. Cameron, Eds., Pergamon Press, Inc., New York, N. Y., pp. 153–189; 1960.
[18] A. Newell, "Learning, generality, and problem solving," *Proc. 1962 Internat'l Conf. on Information Processing*, North-Holland Publishing Company, Amsterdam, The Netherlands; 1963.
[19] A. Samuel, "Appendix: Game of Checkers Played by Mr. R. W. Nealy vs Samuel Checker Playing Program," in "Computers and Thought," E. Feigenbaum and J. Feldman, Eds., McGraw-Hill Book Company, Inc., New York, N. Y.; 1963.
[20] J. McCarthy, Stanford Univ., Stanford, Calif., personal communication.
[21] R. Lindsay, "Inferential Memory as the Basis of Machines which Understand Natural Language," in "Computers and Thought," E. Feigenbaum and J. Feldman, Eds., McGraw-Hill Book Company, Inc., New York, N. Y.; 1963.
[22] B. F. Green, A. Wolf, C. Chomsky, and K. Laughery, "Baseball: an automatic question answerer," *Proc. Western Joint Computer Conf.*, pp. 219–224; 1961.
[23] B. Raphael, "A Computer Representation for Semantic Information," Computation Ctr., Mass. Inst. Tech., Cambridge, Mass., unpublished, 1963.
[24] R. F. Simmons, "Synthex: Toward Computer Synthesis of Human Language Behavior," in "Computer Applications in the Behavioral Sciences," H. Borko, Ed., Prentice-Hall. Inc., Englewood Cliffs, N. J.; 1962.
[25] S. Klein and R. Simmons, "Syntactic Dependence and the Computer Generation of Coherent Discourse," System Development Corp., Santa Monica, Calif., Tech. Memo. TM-758; September, 1962.
[26] J. R. Slagle, "A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus," in "Computers and Thought," E. Feigenbaum and J. Feldman, Eds., McGraw-Hill Book Company, Inc., New York, N. Y.; 1963.
[27] F. Tonge, "A Heuristic Program for Assembly Line Balancing," Prentice-Hall, Inc., Englewood Cliffs, N. J.; 1961.
[28] A. Kuehn and M. Hamburger, "A Heuristic Program for Warehouse Location," Carnegie Inst. Tech., Pittsburgh, Pa., Grad. School of Ind. Admin. Working Paper; 1962.
[29] G. P. E. Clarkson, "Portfolio Selection: A Simulation of Trust Investment," Prentice-Hall, Inc., Englewood Cliffs, N. J.; 1962.
[30] T. C. Evans, "A Heuristic Program for Solving Geometric Analogy Problems," Ph.D. dissertation, Mass. Inst. Tech., Cambridge; 1963 (unpublished).
[31] K. Colby, "Computer Simulation of a Neurotic Process," in "Computer Simulation of Personality," S. S. Tompkins and S. J. Messick, Eds., John Wiley & Sons, Inc.; New York, N. Y.; 1963.

[32] E. A. Feigenbaum, "The simulation of verbal learning behavior," *Prob. Western Joint Computer Conf.*, pp. 121–132; 1961.

[33] E. B. Hunt, "Concept Formation: An Information Processing Problem," John Wiley & Sons, Inc., New York, N. Y.; 1962.

[34] J. Feldman, "Simulation of behavior in the binary choice experiment," *Proc. Western Joint Computer Conf.*, pp. 133–144; 1961.

[35] M. Kochen, "Experimental study of hypothesis formation by computer," *Proc. 4th London Symp. on Information Theory*, C. Cherry, Ed., Butterworths, Washington, D. C.; 1961.

[36] L. J. Fogel, "Toward inductive inference automata," *Proc. 1962 Internat'l Conf. of Information Processing*, North-Holland Publishing Company, Amsterdam, The Netherlands, pp. 395-400; 1963.

[37] H. A. Simon and K. Kotovsky, "Human acquisition of concepts for sequential patterns," *Psych. Rev.*, to be published.

[38] H. A. Ernst, "MH-1, a computer-operated Mechanical hand,' *Proc. 1962 Western Joint Computer Conf.*, National Press, Palo Alto, Calif.; pp. 39–51.

[39] S. Amarel, "On the Automatic Formation of a Computer Program which Represents a Theory, Self Organizing Systems—1962," Spartan Books, Washington, D. C.; 1962.

[40] M. Minsky, "A selected, descriptor-indexed bibliography to the literature on artificial intelligence," IRE TRANS. ON HUMAN FACTORS IN ELECTRONICS, vol. HFE-2, pp. 39–55; March, 1961.

[41] M. Minsky, "A Descriptor-indexed Bibliography to the Literature on Artificial Intelligence," in "Computers and Thought," E. Feigenbaum and J. Feldman, Eds., McGraw-Hill Book Company, New York, N. Y.; 1963.

[42] "Bibliography on Biological and Artificial Intelligence," Jet Propulsion Lab., Calif. Inst. of Tech., Pasadena, Calif., Literature Search No. 254 and Suppl.

[43] P. A. Lachenbruch, A. J. Slevenske, and A. C. Marchese, "Artificial Intelligence—A Summary of Current Research and Development," Am. Inst. for Res., Los Angeles, Calif., No. AIR-C63-2/62-TR; 1962.