

July 1985

Report No. STAN-CS-85-1076
Also numbered KSL-85-38

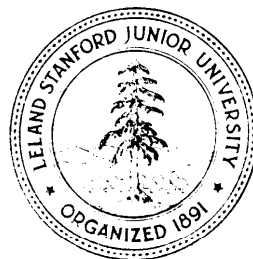
Some Approaches to Knowledge Acquisition

by

Bruce G. Buchanan

Department of Computer Science

Stanford University
Stanford, CA 94305



Knowledge Systems Laboratory
Report No. KSL-85-38

October 1985

Some Approaches to Knowledge Acquisition

Bruce G. Buchanan

Department of Computer Science
Stanford University
Stanford, CA 94305

Some Approaches to Knowledge Acquisition

Bruce G. Buchanan

Stanford University, Department of Computer Science, Stanford, CA, 94305

ABSTRACT

Knowledge acquisition is not a single, monolithic problem for AI. There are many ways to approach the topic in order to understand issues and design useful tools for constructing knowledge-based systems. Several of those approaches are being explored in the Knowledge Systems Laboratory (KSL) at Stanford.

BACKGROUND

In 1969, while working on DENDRAL, we recognized the “bottleneck” problem of acquiring knowledge from experts for use by a knowledge-based system [1]. From that initial recognition, born out of our first efforts at systematizing the process now known as “knowledge engineering”, developed a line of research that is still active at Stanford. In the context of DENDRAL, we first initiated research on interactive editors and automatic rule induction [2]. Then, in the context of MYCIN, we were again confronted with very practical problems of knowledge engineering, and further worked on interactive debugging tools and languages for expressing new knowledge [3].

To date, knowledge engineering has been the only means of building a complex knowledge base, but this remains a tedious process. Thus, we are seeking to develop tools that aid knowledge engineers. Studies in progress include exploring methods by which programs can acquire knowledge by induction from examples, by analogy, by watching, by SOAR’s process of **chunking**, by discovery, and by understanding written text. In this brief overview, we summarize research recently completed or in progress in the KSL. Although we are developing programs in the context of particular, suitable domains, we are seeking methods that are domain independent. Thus some of our research has resulted in papers that analyze and discuss general problems [4, 5, 6, 7].

GENERAL MODEL

Knowledge acquisition cannot be thought of as a single problem; there are several dimensions to the transfer and transformation of problem-solving expertise from a human expert or other knowledge source into a program. In our research, we have identified three different stages of knowledge acquisition and are examining different kinds of learning appropriate to each stage. We have given the chess labels of the “opening,” “middle game,” and “end game” to these stages (see Chapter 5 of [8]). All three can be seen as different perspectives on the general model for learning systems in [4].

In the opening, an expert must lay out the terminology and the problem-solving framework. All subsequent knowledge-acquisition work depends on making this conceptual foundation correct. The middle game builds on the framework that was established initially. In a rule-based system, a specialist provides a large block of rules to cover many cases. In the end game, the knowledge base is refined by applying it to test cases. The TEIRESIAS program [9], is perhaps the best known piece of research on this refinement stage.

The approaches to automating knowledge acquisition described below contribute mainly to the middle- and end-game stages. However, many ideas about the opening stage were recently codified in an experimental system called ROGET [10], which is an EMYCIN-based expert system whose domain is knowledge engineering. ROGET carries on a dialogue with an expert, much as a knowledge engineer does, in order to formulate the working vocabulary and organization for the knowledge base.

Current Work

Knowledge Engineering. We believe it is important for the immediate future to design tools that help experts build knowledge bases for expert systems. These include “smart” editing and debugging tools, as mentioned above, as well as systems that analyze the contents of an emerging knowledge base [11].

One interactive system we are building, called MARCK [12], learns control knowledge for PROTEAN [13]. When an expert is running PROTEAN interactively and overrides PROTEAN’s choice of tasks in favor of another task, MARCK initiates a dialog with the expert to understand the reasons for the override. By examining the differences between the two tasks, MARCK is able to suggest new guidance heuristics that will cause PROTEAN to execute the expert’s preferred task in similar situations in the future.

Another interactive system, called OPAL [14], aids in the definition of new knowledge for the ONCOCIN program, an advisor for management of cancer therapy [15]. OPAL exploits the interactive graphics capabilities of Lisp workstations in order to present a specialist with easily read forms to be filled out. Considerable attention has been paid to human engineering, for example, in the use of menus to minimize the need for typing. Because OPAL has considerable **knowledge** about some types of cancer and about many different drugs, as well as knowledge of treatment plans in general, it can do more than a knowledge engineer who lacks these kinds of details.

Learning from Examples. Induction can be an important method of acquiring new knowledge when libraries of previously solved cases already exist. Because such learning is itself a knowledge-based activity, knowledge about how to learn from examples can be expressed in much the same way as other problem-solving knowledge [2]. We are currently investigating methods of induction in the context of learning rules and **meta-rules** for diagnosing cases of jaundice [16, 17]. The program, called RL, uses a rough model, or half-order theory, of the domain in order to guide a systematic search through a space of plausible concept definitions and associations.

An important problem in theory formation is interpreting observed data in the first place. In the case that an emerging, partially formed theory is used to interpret the data, there is ample opportunity for erroneous extensions to the theory that is being developed to explain the data. We have defined a method for “theory-driven data interpretation” that propagates constraints in order to determine a consistent interpretation of the data. This has been implemented in a program called PRE [18].

Learning by Analogy. Analogical reasoning is not only a powerful problem-solving method but also an important method for constructing knowledge bases. Once one knowledge base is built and refined for an expert system, we should be able to use it as a stepping stone for building other, analogous knowledge bases. There are two main topics to explore: finding appropriate analogies and using those analogies appropriately.

Currently, our focus is on the appropriate use of analogies. We are seeking to acquire problem-solving knowledge in the domain of electrical circuits by exploiting plausible analogies to facts and relations about fluid flow. Given a test problem about a circuit and a statement such as “voltage is like fluid pressure,” the problem solver, called NLAG [19], creates new relations in the electrical domain to solve the problem. Of the many new relations it explores, it will save those that allowed the test problem to be solved.

Learning by Watching. Learning by watching is a form of learning from examples. The learning program is, in effect, an apprentice, watching the problem-solving behavior of a specialist, much as a medical intern observes and interacts with the chief resident. We are investigating the extent to which this learning method can automate the transfer of expertise in building expert systems.

The program we are developing, called ODYSSEUS [20], has several stages of operation. First it must induce the rule and frame knowledge for the system. Using this initial knowledge base as a half-order theory, the program then attempts to infer how the specialist reasons. It does this by enumerating and ranking various plausible reasoning paths that can explain the specialist’s problem-solving behavior. When it can not find a path of reasoning that corresponds to the specialist’s procedure, the program assumes that it is lacking either strategic or domain knowledge. It can attempt to acquire the missing knowledge automatically or by asking specific questions of the expert.

ODYSSEUS can assist the transfer of expertise for both learning and tutoring. In the HERACLES system (a generalized version of NEOMYCIN [21]), it serves as the knowledge

acquisition subsystem by inferring a model of expert behavior. In GUIDON2, the same modeling system is used to infer the model of a novice's behavior.

Learning by Chunking. What we call learning by chunking [22] represents a somewhat different approach to knowledge acquisition than the others described here. Often in the process of solving a problem, we find a way to combine several steps into a single operation, or "chunk," thus improving performance and efficiency. A chunk can be represented as a rule that identifies which aspects of the task were useful in reaching the desired goal. In the context of SOAR, we are developing ways in which a general problem-solving system can use chunking to improve all aspects of its performance.

Learning by Discovery Lenat's thesis on the AM program [23], defines a paradigm of learning by discovery. That program begins with an initial set of basic concepts about a subject area, elementary set theory, and explores the space of new concepts that can be conjectured on the basis of the old ones. Its exploration is guided by numerous heuristics that define the "interestingness" of new concepts formed in various ways from old ones. AM was generalized into the program EURISKO [24].

The Molgen project is currently exploring machine discovery in the context of regulatory molecular genetics. The discovery system will attempt to improve the performance of a qualitative simulation program which predicts the results of experiments in molecular genetics. When incorrect predictions are made the discovery system will suggest modifications to the simulator's theory of molecular genetics. Credit assignment will be accomplished by allowing the discovery system to determine the validity of some of the more observable intermediate states of the simulation, and by giving it predictions for a number of very similar experiments. New terms will be introduced into the theory to account for otherwise unexplainable effects. And, because these generated objects will be described very abstractly when first introduced, the representation of their behavior must be successively refined through the analysis of subsequent experiments. Through subsequent experimentation, the system will modify and verify its emerging theory of gene regulation.

Learning from Written Text. One important source of knowledge is published articles. We have implemented a prototype version of a knowledge acquisition program, called REFEREE [25], that can use this knowledge source more directly. It interacts with an informed (though nonexpert) reader, thus postponing problems of a program's understanding unconstrained English. REFEREE critiques the design and execution of a reported clinical trial on a drug, for example inferring how well the subjects were randomized.

The critique represents a justification for why someone should (or should not) believe the conclusions of the paper. From that justification, then the program can find the new knowledge to be concluded from the study. In the future we will also use the justification to integrate new reports that come from several sources and that may be in conflict with each other.

Current Status

Prototype programs in most of these areas of knowledge acquisition are essentially complete. Active research continues in these approaches, and other approaches are being assessed continually.

Acknowledgements

This work has been funded in part by the following contracts and grants: DARPA N00039-83-C-0136; NIH RR 00785 from the Biomedical Research Technology Program, Division of Research Resources; ONR N00014-79-C-0302; NSF IST 83-12148; and Boeing W266875.

References

- [1] Buchanan, Bruce G., G.L. Sutherland, and E.A. Feigenbaum, "Toward an Understanding of Information Processes of Scientific Inference in the Context of Organic Chemistry,*" in **Machine Intelligence 5**, B. Meltzer and D. Michie, ed., Edinburgh University Press, Edinburgh, 1970.
- [2] Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J. **Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project**. New York: McGraw-Hill, 1980.
- [3] Buchanan, B.G., and E.H. Shortliffe. **Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project**. Reading, MA: Addison-Wesley, 1984.
- [4] Buchanan, B. G., Mitchell, T. M., Smith, R. G. and Johnson, C. R. Jr. "Models of Learning Systems.*" **Encyclopedia of Computer Science and Technology**. 11 (1978) 24-51, also Stanford report STAN-CS-79-692
- [5] Dietterich, T.G. and Buchanan, B.G. "The Role of the Critic in Learning Systems." In **Adaptive Control of Ill-Defined Systems**. Oliver G. Selfridge, Edwina L. Rissland, and Michael A. Arbib, ed., Plenum Press, NATO Advanced Research Institute on Adaptive Control of 111 Defined Systems, June 21-26, 1981, 1984, 127-147.
- [6] Dietterich, Thomas G. and Bruce G. Buchanan "The Role of Experimentation in Theory Formation." In **Proceedings of the International Workshop on Machine Learning, Univ. of Illinois at Urbana-Champaign, June 29-30, 1983**. International Workshop on Machine Learning, June, 1983, 147-155.
- [7] Barr, Avron, Paul R. Cohen and Edward A. Feigenbaum. **The Handbook of Artificial Intelligence, Volumes I, II, and III**. Los Altos, CA: William Kaufmann, Inc., 1981 and 1982.
- [8] Buchanan B.G., et al., "Constructing an Expert System," in **Building Expert Systems**, F. Hayes-Roth, D. Waterman, and D. Lenat, ed., Addison-Wesley, New York, 1983.
- [9] Davis, R., **Applications of Meta-level Knowledge to the Construction, Maintenance, and Use of Large Knowledge Bases**, PhD dissertation, CSD/AI Lab, Stanford University, June 1976, STAN-CS-76-552, HPP-76-7, Tech. Memo 283. Also, reprinted in KBS
- [10] Bennett, J.S. "ROGET: A Knowledge-Based System for Acquiring the Conceptual Structure of a Diagnostic Expert System." **Journal of Automated Reasoning**. 1:1 (1985) 49-74.
- [11] Suwa, Motoi, A. Carlisle Scott, and Edward H. Shortliffe "An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System." **AI Magazine**. 3:4 fall (1982).
- [12] Hayes-Roth, B., and Hewett, M. "Learning Control Heuristics in a Blackboard Environment", Technical report HPP-85-2, Stanford, Ca.: Stanford University, 1985.
- [13] Jardetzky, Oleg, Andrew Lane, Jean-Francois Lefevre, Olivier Lichtarge, Barbara Hayes-Roth, and Bruce Buchanan "Determination of Macromolecular Structure and Dynamics by NMR." In **Proceedings of the NATO Advanced Study Institute, 'NMR in the Life Sciences', June-17-27, 1985**. Plenum Publishing Corp., 1985, forthcoming
- [14] Differding, Joan C. "The OPAL Interface: General Overview"*, Working Paper, Stanford

University Oncocin Projects, August 1984.

- [15] Shortliffe, E. H., Scott, A. C., Bischoff, M. B., Campbell, A. B., van Melle, W., and Jacobs, C. D. "ONCOCTN: An Expert System for Oncology Protocol Management." In **Proceedings of the 7th International Joint Conference on AI**. IJCAI, Vancouver, B C., August, 1981, 876-881.
- [16] Fu, Li-Min, **Learning Object-level and Meta-level Knowledge for Expert Systems**, PhD dissertation, Stanford University, 1985.
- [17] Fu, Li-Min. and Bruce G. Buchanan. "Enhancing Performance of Expert Systems by Automated Discovery of **Meta-Rules**." In **Proceedings of the IEEE Conference, Denver, November** 1984. IEEE, November, 1984.
- [18] Dietterich, Thomas G., **Constraint Propagation Techniques for Theory-Driven Data Interpretation**, PhD dissertation, Stanford University, Dept. of Computer Science, December 1984.
- [19] Greiner, R. and M. Genesereth "The Role of Abstractions in Understanding Analogy", Working Paper **HPP84-8**, Stanford University, Dept. of Computer Science, April 1984.
- [20] Wilkins, D. C., Buchanan, B. G., Clancey, W. J. "Inferring an Expert's Reasoning by Watching." **Proceedings of the 1984 Conference on Intelligent Systems and Machines**. (1984 (also, HPP Report HPP-84-29)) .
- [21] Clancey, W. J. and Letsinger, R. "NEOMYCIN: Reconfiguring a Rule-Based Expert System for Application to Teaching." In **Proceedings of the Seventh International Joint Conference on AI**. TJCAI, Vancouver, British Columbia, 1981, 829-836.
- [22] Rosenbloom, Paul S., John E. Laird, John McDermott, Allen Newell, and Edmund Orciuch "RI-Soar: An Experiment in Knowledge-Intensive Programming in a Problem-Solving Architecture.*" In **Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems**. IEEE, October, 1984.
- [23] Lenat, Douglas B., "AM: Discovery in Mathematics as Heuristic Search," in **Knowledge-Based Systems in Artificial Intelligence**, Randall Davis and Douglas B. Lenat, eds., McGraw-Hill Book Co., New York, 1982, Based on Phd thesis, Stanford University, Stanford, CA, 1977
- [24] Lenat, D. B. "EURTSKO: A Program That Learns New Heuristics and Domain Concepts. The Nature of Heuristics TIT: Program Design and Results." **Artificial Intelligence**. **21** (1983) 61-98.
- [25] Haggerty, John, "REFEREE and RULECRITIC: Two Prototypes for Assessing the Quality of a Medical Paper," Master's thesis, Stanford University, Dept. of Computer Science, 1984.