Experiential Learning in Analogical Problem Solving

Jaime G. Carbonell
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

A computational model of skill acquisition is analyzed based on extensions to an analogical problem solving method and previous AI work on concept acquisition. The present investigation focuses on exploiting and extending the analogical reasoning model to generate useful exemplary solutions to related problems from which more general plans can be induced and refined. Starting with a general analogical inference engine, problem solving experience is, in essence, compiled incrementally into effective procedures that solve various classes of problems in a more reliable and direct manner.[1]

1. Introduction

Whereas humans exhibit a universal ability to learn from experience no matter what the task [14] AI systems are seldom designed to model this adaptive quality. Concept acquisition, i.e. inducing structural descriptions of non-procedural objects from examples, has received substantial attention in the AI literature [9, 7, 11, 17, 18], but with a few exceptions, the techniques developed therein have not been transferred to learning in problem-solving scenarios.[2] Since the process of acquiring and refining problem solving and planning skills is indisputably a central component in human cognition, its investigation from an AI perspective is clearly justified.

In this paper I set out to investigate two hypothesis:

> **Hypothesis:** *Problem solving and learning are inalienable aspects of a unified cognitive mechanism.*

In other words, one cannot acquire the requisite cognitive skills without solving problems --- and, the very process of solving problems provides the information necessary to acquire and tune problem solving skills. The second hypothesis postulates a unified learning mechanism.

> **Hypothesis:** *The same learning mechanisms that account for concept formation in declarative domains, operate in acquiring problem-solving skills and formulating generalized plans.*

One method of testing the second hypothesis is to develop a problem solving mechanism into which one can integrate the techniques developed in concept formation --- with a resultant system that learns from problem solving experience. The analogical problem solving method discussed below provides a framework for automated example generation that enables one to apply learning-from-examples techniques in order to acquire generalized plans. First, I review the basic analogical problem-solving process, and subsequently I discuss the natural incorporation of the experiential learning component.

2. Solving Problems by Analogy

Most human problem solving occurs in a wealth of past experience, whether it be in mundane day-to-day situations, or in the pursuit of skilled tasks, such as structural engineering or medical diagnosis. Hence, past problem solving experience may play a crucial role in both commonplace and expert behavior. The question then becomes: How can a problem solver benefit from experience? That is, how can one solve problems faster, more directly, and with more self assurance simply by having solved similar problems in the past? Or, in more operational terminology: ***How is knowledge of past problem solving behavior recalled and transferred to new problems in similar situations?*** I have addressed the issue of recalling past problem-solving episodes that bear strong similarity to new problems by postulating adaptable, weighted similarity criteria [5, 3] exploiting a memory organization scheme along the lines of Schank's MOPS [16].

Upon recalling similar problem-solving experiences, aspects of the recalled information must be transferred to facilitate and guide the problem solving process. One method of bringing past problem-solving knowledge to bear in new scenarios is based on the analogical problem solving engine [2, 4], summarized below. The operational definition of analogy I adopt involves a *reconstructive mapping*, as evidenced in the following discussion, rather than Winston's more structural approach [19].

Consider a problem space similar to that of GPS [13] or STRIPS [8], in which problem solving occurs by standard Means Ends Analysis,[3] with the added feature that the solution to any given problem must obey a set of path constraints, i.e. global predicates on the sequence of operators that comprise a

[1] This research was sponsored in part by the Office of Naval Research (ONR) under grant number N0014-79-C-0661, and in part by the Defense Advanced Research Projects Agency (DOD), ARPA order number 3597, monitored by the Air Force Avionics Laboratory under contract number F-33615-81-K-1539. The views and conclusions in this document are those of the author and should not interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Agency or the US Government.

[2] The exceptions include Anzai and Simon's Learning-by-Doing Paradigm [1], Mitchell's LEX system [12], STRIPS with MACROPS [8], and Lenat's AM [10].

[3] The reader not familiar with Means Ends Analysis is encouraged to review the technique in any standard AI text, such as Winston's *Artificial Intelligence* [20], in [2], or in the much more thorough treatment in [13].

solution. Now, instead of solving problems in the original problem space in which the states are descriptions of the external world, consider solving the problem by starting with a solution to a similar problem and transforming it into a solution for the new problem. That is, the analogical transformation space consists of states that are themselves complete solutions, and operators that consist of incremental transformations among the set of potential solutions. More explicitly, the analogy transformation space (T-space) is defined by:

- The ***initial state*** is the recalled solution to a past problem that bears strong similarity to the current problem.

- The ***goal state*** is a specification of the solution to the new problem in terms of its original-space initial state, goal state, and path constraints.

- The ***transform operators*** perform all manner of useful edits on a given solution sequence --- such as subgoal-preserving substitutions, splicing in additional steps, deleting redundant steps, performing global parameterization, etc.

- The new T-space ***difference function*** is given by

  $$D_T = \langle D_O(S_{I,1},S_{I,2}),\ D_O(S_{F,1},S_{F,2}),\ D_P(PC_1,PC_2),\ D_A(SOL_1,SOL_2)\rangle$$

  D_O is the difference function between states in the original space. D_P computes differences between path constraints (PC's). D_A measures the applicability of the old solution in the new scenario by determining the fraction of operators in the initial solution sequence (SOL_1) whose preconditions are not satisfied under the new problem specification. S_I denotes an initial state, and S_F denotes a final state. The subscript 1 indexes the retrieved solution, and 2 indexes the specifications on the desired solution to the new problem. D_T is reduced when any of its four components is independently reduced. The problem-solving process in T-space succeeds when $D_T = \langle NIL, NIL, NIL, NIL\rangle$.

- A ***difference table*** indexes T-operators as a function of the remaining T-space differences they reduce between the current solution sequence and the specifications defining the goal state sequence.

Graphically, the MEA Transform Space is depicted in the figure below.

A more detailed discussion of this analogical problem solving method is presented in [2, 4]. Reiterating the basic idea: analogical problem solving proceeds by recalling the solution of a similar problem that worked well, and then transforming that solution to fit the requirements of the new problem at hand.

3. Learning Generalized Plans

The analogical transformation process provides a method of exploiting prior experience in a flexible manner. That is, it requires only that the new problem be structurally similar rather

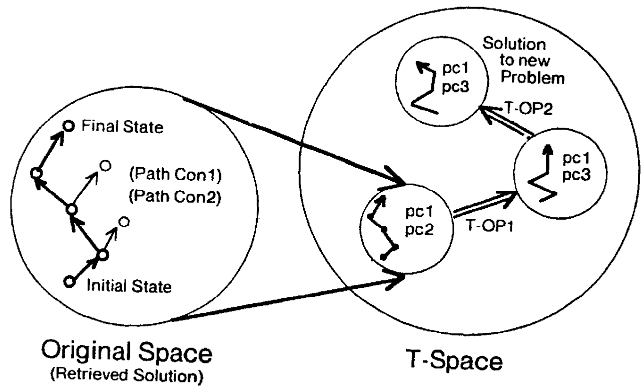

Figure 2-1: A solution path in the original problem space becomes a state in the *analogy transform space.*

than identical to one or more previously solved problems.[4] Hence, simply storing solutions to new problems constitutes a form of learning --- as these can serve as a basis from which solutions to yet newer problems may be analogized. However, there are other aspects to learning that present more interesting challenges. To wit, if a type of problem recurs with sufficient frequency, a human planner is apt to formulate a generalized plan for dealing with future instances of that problem, rather than reasoning analogically from a particular member of that cluster of similar experiences. A generalized plan is, in essence, similar to Schank's notion of a script [15, 6], i.e., a parameterized branching sequence of events with expected goals and default actions.

How is a generalized plan acquired from past problem solving experience? Consider an inductive engine, such as those developed to formulate generalized concepts from sequences of positive and negative exemplars of the target concept [9, 17, 18, 7, 11]. Instead of acquiring disembodied concepts from an external teacher providing training sequences of exemplars labeled "positive" or "negative", in experiential learning the exemplars consist of analogically related past problems and their respective solutions. The concept learned is a generalized plan for solving problems of that type, where the "type" is not artificially defined by an external teacher, but internally defined by clusters of solutions derived analogically from a common ancestor. More specifically:

- Whenever the analogical problem solver generates a solution to a new problem, that solution is tested in the external world. If it works, it becomes a member of the positive exemplar set, together with the prior solution from which it was analogized and other solutions to problems from the same analogical root.

- If the analogized solution fails to work, the cause of the failure is stored and this solution becomes a member of the corresponding negative exemplar set.

---

[4] The MACROPS facility in STRIPS required corresponding initial states and goal states to be identical modulo parameterization of operators in order to reuse portions of past solution sequences [8].

- The positive and negative exemplar sets are given to an induction engine that generates a plan encompassing all the positive solutions and none of the negative exemplars. Thus, past experience solving similar problems provides the training sequence, rather than an external teacher. And, the concept acquired is a generalized solution procedure rather than the description of a static object, as is typically the case in the concept acquisition literature.

- Moreover, negative exemplars are near-misses,[5] since the analogical process generated them by making a small number of changes to known positive instances (i.e., transformations to past solutions of the same general problem type, retaining the bulk of the solution structure invariant). Hence, near-miss analysis can point out the one or two discriminant features between positive and negative exemplars of the general planning structure under construction. In other words, the problem solver serves as an automated example generator, that produces near-misses as a side effect when failing to generate an effective plan.

- The same generalization process used on the solutions can be applied to the problem descriptions corresponding to each solution. Thus, conditions of applicability are generated for each generalized plan.

- Finally, in cases where the analogical problem solver fails to generate a solution for the new problem (as opposed to generating an erroneous solution that becomes a negative exemplar for the generalized plan formation process), different information can be acquired. The situations where a solution was recalled and a plan was formed analogically (independent of whether the plan worked) serve as positive exemplars to reinforce and perhaps generalize the similarity metric used to search memory. The cases where a recalled solution could not be analogized into a candidate plan for the new problem suggest that the old and new problems differed in some crucial aspect not adequately taken into account in the similarity metric, and thus serve as negative exemplars to refine and constrain the similarity criterion.

Thus, we see that analogical problem solving interfaces naturally with a learning-from-examples method in that it provides an internal example generator requiring no external teacher. Presently, I am extending the problem solving engine to extract and use information from the planning process itself (not just problem descriptions and corresponding solutions), such as viable alternatives not chosen, causes of failure to be wary of in similar situations, etc. with a view towards acquiring, or at least refining , the problem solving strategies themselves, in addition to the formation of generalized plans. Parts of the plan generalization process are currently being implemented to test the viability of the proposed knowledge acquisition method; preliminary results are encouraging. Although, much of the theoretical and experimental work in acquiring problem solving skills is still ahead of us, there is sufficient evidence to support the two original hypotheses: the integration of learning and problem solving methods, and the utility of the learning-from-examples technique for acquiring planning skills as well as more static concepts.

[5] Winston [18] defines a *near-miss* as a negative exemplar that differs from positive exemplars in one or two significant features. Near misses are crucial in isolating defining characteristics of a concept in the learning-from-examples paradigm.

References

1. Anzai, Y. and Simon, H. A., "The Theory of Learning by Doing," *Psychological Review*, Vol. 86, 1979 , pp. 124-140.

2. Carbonell, J. G., "A Computational Model of Problem Solving by Analogy," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* August 1981 , pp. 147-152.

3. Carbonell, J. G., "Metaphor: An Inescapable Phenomenon in Natural Language Comprehension," in *Knowledge Representation for Language Processing Systems,* W. Lehnert and M. Ringle, eds., New Jersey: Erlbaum, 1982.

4. Carbonell, J. G., "Learning by Analogy: Formulating and Generalizing Plans from Past Experience," in *Machine Learning,* R. S. Michalski, J. G. Carbonell and T. M. Mitchell, eds., Palo Alto, CA: Tioga Pub. Co., 1982.

5. Carbonell, J. G., "Invariance Hierarchies in Metaphor Interpretation," *Proceedings of the Third Meeting of the Cognitive Science Society,* August 1981 , pp. 292-295.

6. Cullingford, R., *Script Application: Computer Understanding of Newspaper Stories,* PhD dissertation, Yale University, Sept. 1977.

7. Dietterich, T. and Michalski, R., "Inductive Learning of Structural Descriptions," *Artificial Intelligence,* Vol. 16, 1981 .

8. Fikes, R. E. and Nilsson, N. J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence,* Vol. 2, 1971 , pp. 189-208.

9. Hayes-Roth, F. and McDermott, J., "Knowledge Acquisition from Structural Descriptions," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* 1977 , pp. 356-362.

10. Lenat, D., *AM: Discovery in Mathematics as Heuristic Search,* PhD dissertation, Stanford University, 1977.

11. Mitchell, T. M., *Version Spaces: An Approach to Concept Learning,* PhD dissertation, Stanford University, December 1978.

12. Mitchell, T. M., Utgoff, P. E. and Banerji, R. B., "Learning Problem-Solving Heuristics by Experimentation," in *Machine Learning,* R. S. Michalski, J. G. Carbonell and T. M. Mitchell, eds., Palo Alto, CA: Tioga Pub. Co., 1982.

13. Newell, A. and Simon, H. A., *Human Problem Solving,* New

Jersey: Prentice-Hall, 1972.

14. Newell, A. and Rosenbloom, P., "Mechanisms of Skill Acquisition and the Law of Practice," in *Cognitive Skills and Their Acquisition,* J. R. Anderson, ed., Hillsdale, NJ: Erlbaum Assoc., 1981.

15. Schank, R. C. and Abelson, R. P., *Scripts, Goals, Plans and Understanding,* Hillside, NJ: Lawrence Erlbaum, 1977.

16. Schank, R. C., "Language and Memory," *Cognitive Science,* Vol. 4, No. 3, 1980 , pp. 243-284.

17. Vere, "Inductive Learning of Relational Productions," in *Pattern-Directed Inference Systems,* Waterman and Hayes-Roth, 1978, eds., New York: Academic Press, 1978.

18. Winston, P., *Learning Structural Descriptions from Examples,* PhD dissertation, MIT, September 1970.

19. Winston, P. H., "Learning and Reasoning by Analogy," *CACM,* Vol. 23, No. 12, 1979 , pp. 689-703.

20. Winston, P., *Artificial Intelligence,* Reading, MA: Addison Wesley, 1977.