Concept Formation over Explanations and Problem-Solving Experience *

Jungsoon Yoo Doug Fisher
Box 67, Station B, Department of Computer Science
Vanderbilt University, Nashville, TN 37235 U. S. A.
yoojp@vuse.vanderbilt.edu dfisher@vuse.vanderbilt.edu

Abstract

Recent research suggests the utility of performing induction over explanations. This process identifies commonalities across explanations that cannot be extracted solely by explanation-based techniques. This has important implications for the 'correctness' of learned knowledge [Flann and Dietterich, 1989] and, as we show, on the efficiency with which learned knowledge can be reused. Specifically, we illustrate that inductive *concept formation* can abstract and organize explanatory knowledge for efficient reuse in a domain of algebra story problems.

1 Introduction: Principles of Memory Organization

An increasingly well-accepted view in psychology and AI is that problem solving is a process of classification. Performance improves by learning patterns that discriminate between problem-solving choices. Production system models of learning and problem solving (e.g., [Langley, 1985]) make the 'problem solving as categorization' view explicit, as do models of case-based reasoning [Kolodner, 1987]. Another paradigm concerned with learning and problem solving is explanation-based learning [Mitchell *et at,* 1986; Minton, 1988]. Each of these three strategies aims to improve the efficiency with which knowledge is accessed and reused in novel problem-solving situations.

Of these three approaches to improving problem-solving performance, we will focus on explanation-based learning, though the general approach that we describe is relevant to other paradigms as well. In particular, we have looked at explanation-based learning as a model of learning to solve *algebra story problems* IMayer, 1981]. Consider the problem,

> *"A train leaves a station and travels east at 12 km/h. Three hours later a second train leaves and travels east at 120 km/h. How long will it take to overtake the first train?"*

*This research was supported by NASA Ames grant NCC 2-645. We thank the reviewers for useful comments.

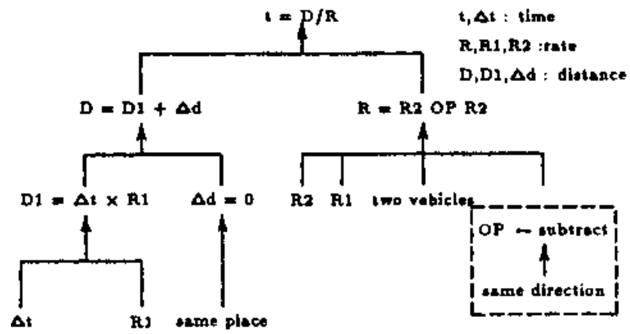

Figure 1: A generalized problem solution trace (explanation).

A learner with a set of primitive rules that constitute domain-specific knowledge (e.g., Distance = Rate * Time) and general algebraic knowledge (e.g., a*b = c $\rightarrow a = c/b$) forms a solution to this *overtake* problem that is abbreviated in Figure 1. The time until overtake (Time = Distance / Rate) may be obtained from the distance that must be made up by the faster train ($D = D1 + \Delta d$, where D1 is the distance traveled by the first train before the second train starts), and the relative rate of travel of the second train (R = R2 — RI). Explanation-based learning will generalize the arguments (i.e., by turning them to variables in a controlled way) of this AND-tree solution trace so that it can be reused on future problems.

Unfortunately, even after variablization, the applicability of this solution trace will be highly limited. Consider a second *opposite-direction* problem,

> *"Two trains leave the same station at the same time. They travel in opposite directions. One train travels 64 km/h and the other 104 km/h. In how many hours will they be 1008 km apart?"*

which has a solution structure almost identical in form to the solution of Figure 1; it differs only in the structure of the boxed subtree. Nonetheless, the earlier solution cannot be used to solve the new problem unless the basic EBG process is altered. Flann and Dietterich [1989] and others [Hirsh, 1988; Pazzani, 1988] suggest the utility of performing *Induction over Explanations* (IOE) which can generalize an explanation beyond simple variablizar tion. IOE does this by superimposing a set of AND-tree explanations, and pruning subtree structures that are not shared by all trees. For example, performing IOE over the explanation trees for an opposite-direction and overtake problems would yield a tree structure with the boxed rightmost subtree of Figure 1 severed.

Using IOE, the common substructure of multiple explanations can be reused in a wider variety of situations. The substructure does not provide the complete explanation, but it ideally provides a sizable chunk that can be completed using the primitive rules of the original domain theory. Thus, induction over explanations abstracts redundant substructures out, thus promising to improve the efficiency with which applicable learned knowledge can be found and reused.However, unconstrained induction can remove all the benefits of explanation-based learning. Consider that there may be radically different explanations of why oat-bran cereal and baked fish are health-food. Abstracting out the common substructure might yield an 'explanatory' substructure, health-food(x), which is a trivial statement of the target concept. Thus, on one hand maximally-operational explanations provide a complete explanation for new situations, but it is difficult to find applicable explanations since redundant structures must be searched multiple times. Conversely, maximally-general structures *(e.g.,* cup(x), health-food(x)) make it easy to find applicable past 'experience', but nothing useful is gleaned in having done so.

The tradeoffs of an explanation-based memory are closely related to tradeoffs traditionally found in inductive concept formation [Fisher, 1987; Lebowitz, 1982; Kolodner, 1987]. Ideal object concepts are those where many features are *predictable (i.e.,* favoring specific concepts), but many features are also *predictive (i.e.,* favoring general concepts). Our concern with 'prediction' accuracy in explanation-based learning may seem at odds with the traditional view that *efficiency* is the critical performance dimension in explanation-based learning. However, choices in a domain theory search constitute predictions; informed and accurate decisions at choice points result in an efficient search, while erroneous decisions are the cause of backtracking and inefficiency. The hypothetical curve of Figure 2 illustrates performance trends that might be expected. Maximizing predictiveness (generality) will underfit the data, necessitating uninformed prediction *(e.g.,* uninformed search through a domain theory). Conversely, maximizing predictability (operationality) will overfit the data and introduce redundancies into the search for applicable past experience. An ideal abstraction should be relatively unique and easily identified; it can then predict a sizable portion of the complete explanation structure; the domain theory is only required to complete the proof *(e.g.,* of relative rate).

This paper shows that inductive methods of *concept formation* [Gennari *et al,* 1989] can abstract redundancy out of explanations and organize shared substructure(S) to improve the efficiency of finding explanations

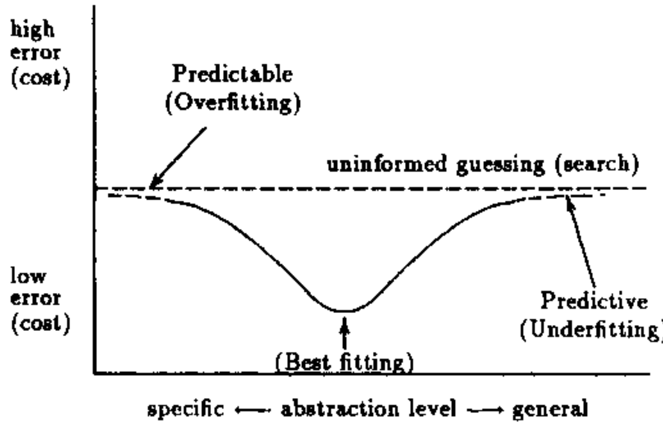

Figure 2: Prediction error (and search efficiency).

for reuse. Section 2 describes our system, EXOR (Explanation ORganizer), which clusters and classifies explanations based on shared structure. In section 3 we report experimental results which illustrate that ExOR effectively improves problem-solving efficiency in a domain of algebra story problems. Sections 4 and 5 analyzes the strengths and weaknesses of the approach, and details its relation to ongoing research in explanation-based learning.

2 Concept Formation over Explanations

Our system, EXOR, performs concept formation over explanations. We use the domain of algebra story problems [Mayer, 1981] to describe and evaluate our system. In particular, EXOR embeds IOE within a control structure for building abstraction hierarchies that was inspired by Lebowitz's [1982] UNIMEM and Fisher's [1987] COBWEB. Figure 3 gives an example of the type of abstraction hierarchy formed by EXOR over algebra story problems that span 16 types *(e.g.,* overtake, opposite direction, roundtrip) taken from Mayer [Mayer, 1981]. Solutions to these problems range from very similar to quite different. The domain theory includes formulae like those described above with a variety of ways for solving the quantities: Distance, Time, and Rate. Within each node of the abstraction hierarchy is a generalized-explanation subtree that is common to all descendents of the node. If there is no common substructure over the entire set of observed explanations, then the root of the hierarchy will be empty.

To incorporate an explanation into a classification tree, the explanation is compared to the explanation substructure of a node of the abstraction hierarchy (initially the root) and the remainder of the IOE procedure is applied to generalize the new explanation and the node's substructure. If this results in a generalization that is equivalent to the node's generalized explanation, then the new explanation must be more specific than the node's partial explanation; in this case classification of the explanation proceeds to the children of the node.

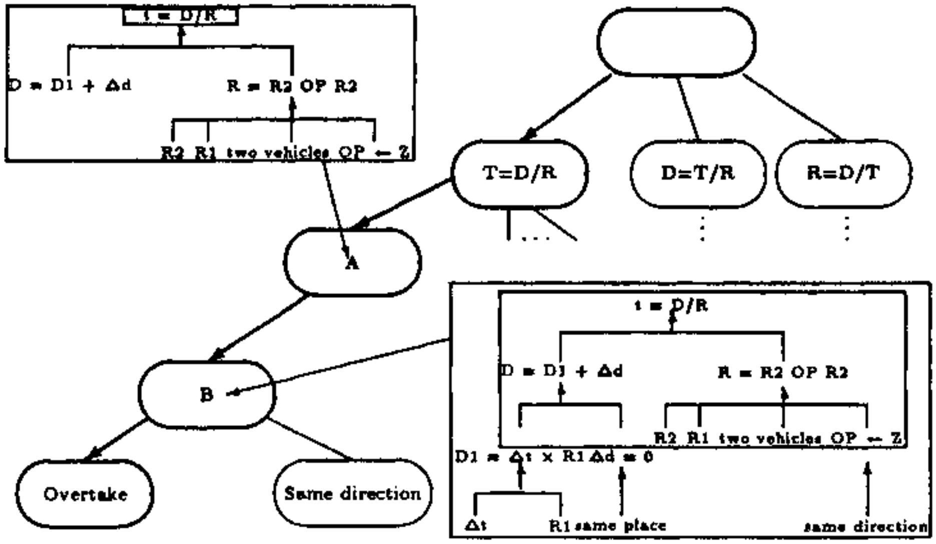

Figure 3: A classification tree over explanations (problem solving traces).

If IOE yields a structure that is more general than the current node, then the node's explanation structure is not more general than the new explanation; in this case, the new explanation is made a sibling of the node. Of course, to be useful the hierarchy must not simply be used to store explanations, but it should facilitate explanation construction. In this case, EXOR solves a target concept *(e.g.,* distance) using a problem statement of operational predicates and the classification tree. If the subexplanation stored at the node is applicable to the new problem and the variable instantiation constraints (if any exist) can be satisfied by the new problem then one of the node's children is selected for investigation.

Figure 4 illustrates the basic classification process. As search down the classification tree proceeds, EXOR extends the solution to the current problem using the subexplanations that are stored at each node along the path. If a contradiction occurs between a node's partial explanation and the known conditions of the problem statement, then the node is abandoned, its conditions (the partial explanation) are retracted, and search control looks to a sibling of the node. That is, control returns to the node's parent and another child is explored. If all of a node's children result in contradictions *then an attempt is made to complete the partial explanation accumulated thus far by using the domain theory.* This 'last resort' is represented in Figure 4 by the 'house-shaped', dashed boxes that emanate from examined nodes. If this fails then the node is abandoned as above (i.e., its conditions are retracted and backtracking returns control to the node's parent).

Figure 4 indicates that classification is not necessarily deterministic. Some search of the tree (and domain theory) is still required, though we hope that this is less than an uninformed search of the domain theory. To direct classification, a measure of *category utility* [Gluck and Corter, 1985] is used to rank the promise of children under a node. In addition to storing a partial explanation at a node, which is true of *all* of the node's descendents, we store statistics on the distribution of operational predicates of explanations stored under the node - statistical trends in operational predicates can be used to heuristically guide the selection of nodes from which EXOR builds an explanation for the current problem.

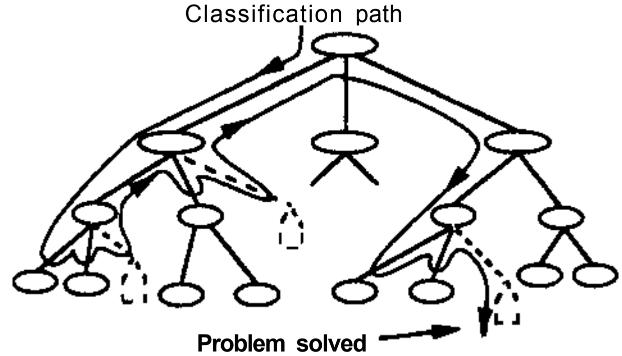

Figure 4: Problem solving by classification.

Intuitively, category utility is a measure of the predictability and predictiveness of a new problem's operational predicates relative to a category - i.e., a classification tree node. The predictability of a predicate F_k relative to a category (node) N_i is given as $P(F_k|N_i)$: the probability that F_k will participate in an explanation stored under N_i. The predictiveness of a predicate is given by $P(N_i|F_k)$: the probability that an explanation with F_k will be stored under N_i. Recalling the discussion from Section 1, category utility is a tradeoff between these two factors: $\sum_k P(F_k)P(N_i|F_k)P(F_k|N_i)$, where $P(F_k)$ weights the importance of the tradeoff for the most frequently observed predicates. When a problem statement is presented to EXOR, candidate nodes are ranked by their category utility scores over the operational predicates. The highest scoring nodes are investigated first. Contradictions may still arise, causing EXOR to abandon a proposed node, but the inductive assumption is that the distribution of operational predicates provides considerable heuristic guidance.

3 Experimental Results

EXOR'S ability to improve problem-solving efficiency in the domain of 48 algebra problems was tested. A subset of 32 problems were selected for training and 16 were selected for testing. At intermittent points in training (*i.e.,* every four problems), the performance of the EXOR classification tree was evaluated. In particular, we compared the total number of predicates instantiated using a domain theory search (no learning), an EBG-like system, and the heuristically-guided ExOR tree search over the 16 test problems.[1]

Figure 5 illustrates the total number of predicates instantiated during a domain theory search and an EXOR classification tree search as training proceeds over the 32 training problems. The experiment has been run ten times with different random orderings of the data. The dashed horizontal line reflects the total work performed from domain theory search alone (*i.e.,* no learning) over the 16 test problems. The amount of work performed by EXOR trees is also graphed, but recall from the description of the explanation-construction procedure that search using an EXOR classification tree stems from two sources. First, EXOR searches a path in the classification tree to find a maximally-specific node that appears to be applicable to the new problem. The partial explanation at such a maximally-specific node may not contradict the operational (observed) predicates of the new problem, but the partial explanation may not be successfully extended by any of the node's children. However, before a node is abandoned, a final effort is made to extend the partial explanation using the domain theory; these (nested) domain theory searches are the second source of search. The shaded (lower, increasing) area reflects search in tree nodes; the upper (decreasing) curve gives

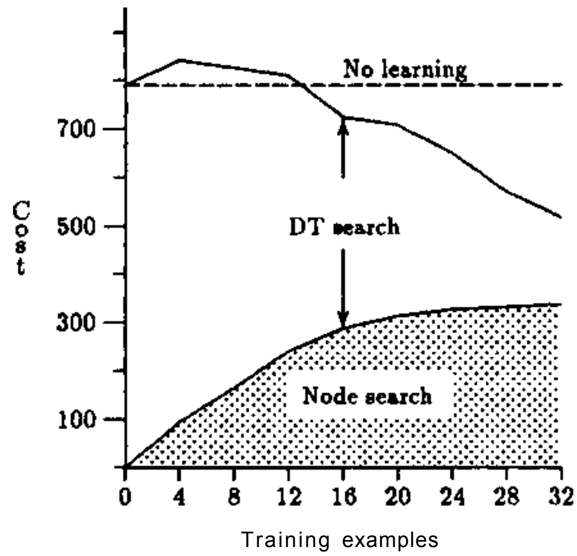

Figure 5: Performance as a function of training.

the total amount of search required to solve all 16 test problems including the domain theory search (i.e., the difference between the upper and lower curves) to complete partial solutions. Thus, EXOR reduces the overall effort required to solve problems (i.e., the decreasing curve); the effort that is required is increasingly borne by the ExOR classification tree, while the domain theory plays a corresponding smaller role as training proceeds.

The efficiency of EBG has not been graphed, but it was tested on the same data. After 32 training problems, EBG required 1352.6 predicate instantiations to solve 16 test problems. This is considerably more than either the domain theory alone or EXOR. EXOR'S relative success stems from its ability to exploit shared partial solutions. For example, EXOR can exploit the generalized solution from an 'opposite direction' and 'overtake' problem to partially solve a 'closure' problem; something that EBG can not do. This limitation of EBG is magnified when there are many explanations that differ in very minor ways. For example, suppose that an opposite-direction problem describes a car traveling east and the other traveling west, and there are domain theory inference rules that tell us that east and west are opposite-directions, as are north and south. EBG will not be able to exploit the solution to the first problem in its attempt to solve a new problem, which is identical to the first, except that the cars are traveling north and south. In fact, the great similarity between problems may lead to a considerable amount of redundant search until a contradiction is found. This was the case with many of the problems in our domain, thus the poor performance of EBG. However, EBG proves quite adequate on problems that structurally match previously-observed problems. In the following section we will discuss this and other issues of explanation-based learning, and the manner in which these limitations are addressed by EXOR using lessons adapted from inductive learning.

[1] There are many dimensions over which we could have compared performance (and did): the total number of rules examined, and the total number of backtracks. We report predicate instantiations, as this is the most granular dimension. We did not choose to systematically investigate cpu time since the 'no learning' condition and our EBG implementation take direct advantage of the Prolog interpreter in applying rules, while EXOR includes some overhead that is implementation-based, and not of theoretical consequence. This is not to say that all of our system's overhead is simply implementation-based. For example, the use of category utility and probabilistic matching are legitimate theoretical concerns in terms of rule match cost. Future work will have to untangle theoretical from implementation overhead. Suffice it to say that currently EXOR and EBG appear comparable in terms of cpu time, though this is a dubious comparison in favor of EBG. See Segre, Elkan, and Russell [1990] for a good summary of possible performance dimensions.

4 Selective Utilization and Pruning

In addition to our experimental demonstrations, EXOR classification hierarchies also suggest natural approaches to specific issues in explanation-based learning. One of these concerns the *selective utilization* [Markovitch and Scott, 1989; Mooney, 1989] of learned knowledge. Under what conditions should learned knowledge be exploited and when is it best to rely solely on the initial domain theory? Markovitch and Scott's LASSY accumulates statistics on how frequently each antecedent of a rule will fail; learned rules are only used in attempts to prove antecedents that have succeeded sufficiently often *(e.g.,* at least 50% of the time). The justification for this strategy is that if a subgoal is likely to fail then one should not search for a subproof in vain *twice* - once with learned rules and once with the domain theory from which the learned rules were constructed. Mooney's [1989] EGGS uses a technique that is similarly motivated. These approaches mitigate the utility problem and improve efficiency, but nonetheless suffer from two limitations. First, the likelihood of subgoal failures is only estimated within the context of a single (domain theory) rule; intuitively, one might expect that the likelihood of a subgoal failure would be dependent on the more complete problem solving context. Second, LASSY and EGGS decide to make all learned rules available for examination or none; rather we believe that the relevance of learned rules will vary with problem solving context. It should be possible to ignore rules that are deemed irrelevant.

An EXOR classification hierarchy addresses both limitations. Each node represents the status of the complete problem-solving context. The system maintains statistics much like LASSY's number of backtracks (failures) at each node. Nodes with an unacceptable number of backtracks *(e.g.,* greater than 50% of the time) are *pruned* [Quinlan, 1986]. If all of a node's children are pruned then this effectively identifies the problem solving contexts in which the system should rely exclusively on domain theory. Those children that do remain serve to identify learned extensions that have been previously applicable; learned rules that are not present in these children are not considered when attempting to extend the explanation from the current node. Figure 6 illustrates the classification process after pruning low utility nodes. Rather than overfit the problem solution, EXOR turns to the domain theory at 'suitable' levels of specificity.

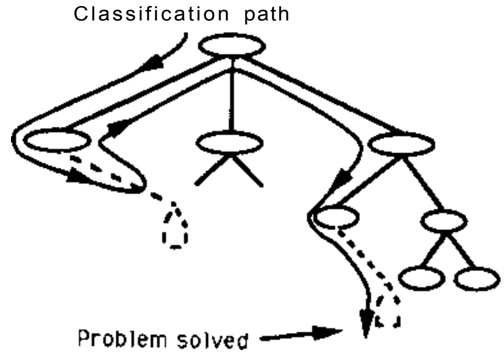

Figure 6: Classification after pruning low utility nodes.

5 Cost-Effective Features

In addition to pruning, we have also investigated a second extension. Initially, EXOR'S search procedure was guided by a category utility score computed solely over predicates that are known to be true from the problem statement. Thus, problem statements draw an initial *boundary of operationality* [Braverman and Russell, 1988]. However, this boundary is not necessarily optimal for purposes of efficiency. For example, recall from an earlier example that predicates such as east and vest convey little information *per $e* - it is the inferred predicate, opposite-direction, that distinguishes solutions to which a problem corresponds. Thus, improved performance is expected by combining some forward-chaining capabilities *(e.g.,* inference from the problem statement to an appropriate boundary) with the backward chaining mechanisms that currently dominate EXOR'S processing.

The success of forward chaining depends on identifying predicates *(e.g.,* opposite-direction) that differentiate categories of different problem solving experiences, thus better focusing the search for solutions to new problems. However, proving or disproving the truth of a predicate requires effort as well. Thus, an ideal boundary of operationally includes predicates with greater efficiency benefits than costs. We can formalize this notion in terms of the *expected number* of problem-solving steps (or predicates instantiated, or any of several other measures of *cost)* required to solve a problem with and without knowledge of a predicate's truth. Let *E(c\N)* be the expected cost of solving an arbitrary problem beginning at node *N* of an EXOR classification tree. Assume that we investigate the children, C,-, of TV in order of probability. Our inductive assumption is that children will successfully extend the current problem with roughly the same probability that they successfully classified earlier problems. Thus,

$$
\begin{aligned}
&E(c|N)\\
&= P(C_{max})[E(c|C_{max})] +\\
&\quad P(C_{max-1})[E(c|C_{max-1}) + U(c|C_{max})] + \ldots +\\
&\quad P(C_{max-m})[\,E(c|C_{max-m})\\
&\qquad\qquad + U(c|C_{max}) + \ldots + U(c|C_{max-m+1})]
\end{aligned}
$$

where $P(C_{max}) \geq P(C_{max-1}) \geq \ldots \geq P(C_{max-m})$, $E(c|C_j)$ is the expected cost *(e.g.,* number of steps) of successfully finding the problem's solution under a child, *Cj*, and $U(c|C_j)$ is the expected cost in an unsuccessful search of *Cj* for a solution to the problem. Thus, the cost of finding a solution in the second most probable subtree of *N*, C_{max-1}, includes the cost of having first searched the most probable node unsuccessfully. These quantities can be computed or at least approximated from the statistics that are maintained in the tree *(e.g., P(Cj))* and from the structure of the tree itself (e.g.,

Table 1: Performance comparison of EXOR with different options.

	original EXOR	/w pruning and forward chaining	improvement (%)
tree search	337.0	316.4	6.1%
DT search	180.9	140.6	22.3 %
total search	517.9	457.0	11.8 %

$E(c|C_j)$). Using similar constructions we can approximate the expected cost when the truth of a predicate F_k is known, $E(c|N, F_k)$, and the expectation in the case of $\neg F_k$, $E(c|N, \neg F_k)$.

Putting these quantities together, it is useful to forward chain in an attempt to verify a predicate if

$$E(c|N) > P(F_k|N)[E(c|N, F_k) + E(cp\ F_k)] + [1 - P(F_k|N)][E(c|N, \neg F_k) + E(cp\ \neg F_k)],$$

where $E(cp\ F_k)$ is the expected cost to prove the predicate F_k, $P(F_k|N)$ is the probability that we will be able to prove the truth of F_k and $1 - P(F_k|N)$ is the probability that it is not true. In general, knowing the complement to be true can also be predictive of a particular course of action, and ExOR exploits this knowledge as well. $P(F_k|N)$ can be approximated by the proportion of explanations stored under N] that contain F_k and the proportion that do not. $E(cp\ F_k)$ and $E(cp - F_k)$ are currently approximated from the domain theory: how many rules contain F_k as an consequent and how many rule combinations can conclude F_k - assuming that the combinations that are responsible for concluding F_k are equiprobable, the expected cost is proportional to approximately 1/2 the cardinality of this set. Intuitively, the greater the branching factor and distance of F_k from the initial operational (problem statement) boundary, the greater the cost of inferring it.[2]

Notice that the utility of forward chaining on a particular predicate F_k varies with the problem-solving context (i.e., node N); we only expend work on inferring F_k when it will help distinguish N'S descendents. Once identified, statistics on these predicates are used in the category utility calculation to bias search in the most promising directions.

Table 1 shows the result of the EXOR'S performance after the 32 training examples with forward chaining and pruning. These extensions reduce the search within the EXOR tree by 6.1% and the saving from the domain theory search to complete a partial solution is 22.3%. The total savings from forward chaining and pruning is 11.8%.

6 Concluding Remarks

EXOR abstracts redundant explanation substructures and organizes them hierarchically for reuse, thus yielding advantages in terms of problem-solving efficiency.

[2]Notice that $E(cp\ F_k)$ is not conditioned on N, though this is clearly a preferable strategy; explanations under N may exhibit a relatively small number of combinations to prove N. Thus, we wish to approximate $E(cp\ F_k|N)$ in the future.

More generally, our research seeks to unify principles of inductive and explanation-based learning. First, traditional explanation-based concerns with efficiency can be cast in terms of the traditional inductive performance dimension of prediction accuracy: accurate prediction along search choice points result in a more efficient search [Carlson *et al.*, 1990; Fisher and Chan, 1990]. To some extent this relationship was recognized in earlier work that treated search-control learning as a problem of concept induction [Mitchell *et al.*, 1983; Langley, 1985]. However, we have strengthened this connection in several ways. Notably, principles of feature predictiveness and predictability, which play a considerable role in inductive concept formation systems [Lebowitz, 1982; Kolodner, 1987; Fisher, 1987], are also used to identify informative, cost-effective predicates and to guide the search for relevant past experience with these predicates. Second, too much emphasis on feature predictability (specificity) can lead to *data overfitting* in explanation-based learning, as well as in inductive systems [Quinlan, 1986; Fisher and Chan, 1990] where it has been a long recognized problem. Pruning in EBL contexts, as with inductive systems, mitigates the problem. Finally, EXOR'S inductively-motivated approach addresses some specific research concerns in EBL, notably the problem of selective utilization [Mooney, 1989; Markovitch and Scott, 1989] and identifying appropriate boundaries of operationality [Braverman and Russell, 1988].

A second research direction is to extend EXOR to other domains, particularly fault diagnosis. Many engineering projects (*e.g.,* designing a purifier/pump system) construct a *fault tree* [Malasky, 1982], which is an AND/OR structure that describes the events (singly and in combination) that may lead to a top-level fault (*e.g.,* loss of pump flow). Search for causes in this AND/OR space is analogous to a domain theory search, and is thus amenable to speedup. As a human-engineered artifact however, there are often inconsistencies in the fault tree. Thus, we plan to use the fault tree as an initial domain theory, but to use ExOR to organize experience and more efficiently guide diagnosis; logical inconsistencies may remain or they may be 'pruned' out, but in any case explanation patterns that better reflect the system's true behavior will come to statistically dominate EXOR'S reasoning. Thus, this approach to inconsistent domain theories is similar in intent to systems like Towell, Shavlik, and Noordewier's [Towell *et al.*, 1990] neural net/EBL system, albeit with very different approaches to the inductive learning component.

References

[Braverman and Russell, 1988] Braverman, M. and Russell, S. Boundaries of operationally. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 221-234). Ann Arbor, MI: Morgan Kaufmann, 1988.

[Carlson *et ai,* 1990] Carlson, B., Weinberg, J., and Fisher, D. Search control, utility, and concept induction. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 85-92). Austin, NY: Morgan Kaufmann, 1990.

[Fisher, 1987] Fisher, D. H. Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2,* 139-172, 1987.

[Fisher and Chan, 1990] Fisher, D. H. and Chan, P. Statistical Guidance in Symbolic Learning. *Annals of Mathematics and Artificial Intelligence,* 1990.

[Flann and Dietterich, 1989] Flann, N. and Dietterich, T. A Study of Explanation-Based Methods for Inductive Learning. *Machine Learning, 4,* 187-226, 1989

[Gennari *et ai,* 1989] Gennari, J., Langley, P., and Fisher, D. Models of incremental concept formation. *Artificial Intelligence, 40,* 11-62, 1989.

[Gluck and Corter, 1985] Gluck, M. and Corter, J. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society* (pp. 283-287). Irvine, CA: Lawrence Erlbaum, 1985.

[Hirsh, 1988] Hirsh, H. Knowledge as bias. *Proceedings of the First International Workshop in Change of Representation and Inducttve Bias,* 186-192, 1988.

[Kolodner, 1987] Kolodner, J. L. Extending problem solver capabilities through case-based inference. *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 167-178). Irvine, CA: Morgan Kaufmann.

[Langley, 1985] Langley, P. Learning to search: from weak methods to domain-specific heuristics. *Cognitive Science, 9,* 217-260, 1985.

[Lebowitz, 1982] Lebowitz, M. Correcting erroneous generalizations. *Cognition and Brain Theory, 5,* 367-381, 1982.

[Malasky, 1982] Malasky, S. *System safety: Technology and application* Garland STPM Press, New York, 1982.

[Markovitch and Scott, 1989] Markovitch S. and Scott P.D, Information Filters and Their Implementation in the SYLLOG System *Proceedings of the Sixth International Workshop on Machine Learning* (pp. 404-407). Ithaca, NY: Morgan Kaufmann, 1989.

[Mayer, 1981] Mayer, R. Frequency norms and structural analysis of algebra story problems into families, categories, and templates. *Instructional Science, 10* (pp. 135-175), 1981.

[Minton, 1988] Minton, S. Quantitative results concerning the utility of explanation-based learning. *Proceedings of the Seventh National Conference on Artificial Intelligence* (pp. 564-569). St. Paul, MN: Morgan Kaufmann, 1988.

[Mitchell *et at,* 1986] Mitchell, T., Keller, R., and Kedar-Cabelli, S. Explanation-based learning: a unifying view. *Machine Learning, J,* 47-80, 1986.

[Mitchell *et at,* 1983] Mitchell, T., Utgoff, P., and Banerji, R. Learning problem solving heuristics be experimentation. In R. Michalski, T. Mitchell, and J. Carbonell (Eds.) *Machine Learning: An Artificial Intelligence Approach,* Palo Alto, CA: Morgan Kaufmann, 1983.

[Mooney, 1989] Mooney, R. The effect of rule use on the utility of explanation-based learning. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 725-730).Detroit, MI: Morgan Kaufmann, 1989.

[Pazzani, 1988] Pazzani, M.J. *Learning causal relationships: an integration of empirical and explanational based learning methods.* Ph.D. Thesis, University of California, Los Angeles, 1988.

[Quinlan, 1986] Quinlan, J. R. Induction of decision trees. *Machine Learning, 1,* 81-106, 1986.

[Segre *et al,* 1990] Segre, A., Elkan, C, and Russell, A. A critical look at experimental evaluations of EBL. *Machine Learning, 6,* 1991.

[Towell *et al,* 1990] Towell, G., Shavlik, J., and Noordewier, M. Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 861-866). Boston, MA: Morgan Kaufmann, 1990.