Introduction

How can teachers tell who is the brightest kid in the class? Even before the first test results are in, they will have a good idea who the gifted students are by their performance in class. Not only do they answer questions well, but they tend to pose questions frequently and skillfully. By asking questions, good students take greater advantage of their best resource, namely the teacher. Active learners are curious.

Systems for machine learning can also benefit from curiosity. A cardinal advantage which is gained by enabling systems to formulate examples or test cases on its own is that the properties of the examples can be controlled. It is well known that the manner in which examples are selected can effect the style and efficiency of learning (eg Rissland, 1988). In many cases the order of presentation of examples determines whether a concept can be learned or not (eg Gold, 1967); in others (eg version space) it does not govern learnability as such but does have an enormous influence on the amount of searching required.

A striking defect of current machine learning systems is that their teacher must be well aware of the internal representations used to guide them towards successful performance — which makes interacting with them more like programming than teaching (Witten & MacDonald, 1987). For example, the notion of "near miss" is relative to the internal representation used: a near miss for one representation might be a far miss for another, and *vice versa.* Question-asking helps to decouple the teacher from the internals of the learning system. The machine is in the best position to determine what might constitute a near miss or near hit relative to the representation it is using.

For people, one important result of transferring initiative from teacher to learner is that it shifts responsibility for understanding what the example "means" in the current context from learner to teacher. When the teacher chooses an example, the learner may be at a loss to interpret what is meant by it. Being more knowledgeable, the teacher is presumably in a better position to interpret the student's chosen example, perhaps using it to pinpoint errors in his conception. Analogously, in the machine learning situation a teacher-chosen example may only be interpretable at the cost of an inordinate amount of search. Conversely, a system-chosen example may exhibit a misconception that needs to be corrected. However, it is difficult to exploit this in machine learning, for without the rich texture of natural-language communication we cannot move up to the meta-level to discuss the appropriateness (or otherwise) of an example.

A final potential advantage of student-directed dialogue is that it admits the possibility of challenging the accuracy of input, perhaps allowing mistakes made by the teacher to be corrected. Again, this is not exploitable by current systems since they do not represent or reason about the veracity of their sources of information (other than perhaps on a purely statistical basis).

Question-asking is likely to prove to be a powerful technique in both similarity- and explanation-based learning. For similarity-based learners, questions can be used to

S 1a. expose inadequacies in the representation language (eg inappropriate bias, Utgoff, 1986)
S 1b. challenge the accuracy of the information given
S 2. assist with the testing of hypotheses
S 3. debug concepts by asking questions to track down apparent inconsistencies caused by over- and under-generalizations (eg Shapiro, 1983; Lloyd, 1986).

For explanation-based learners, questions can be used to

E 1. clarify goals
E 2. assist in completing solution traces, either to compensate for a weak domain theory (Pazzani, 1987) or to expedite the proof process (DeJong & Mooney, 1986).
E 3. help to define operationality criteria on the fly
E 4. validate generalizations before storing them as part of the domain theory.

The thumbnail sketches of similarity- and explanation-based learning in Figure 1 indicate where each of these opportunities arise.

This paper addresses the issue of hypothesis-testing for similarity-based learners only (S 2 above), although it is relevant to the problem of validating generalizations for explanation-based learners (E 4). A similarity-based learner can formulate alternative hypotheses about the concept to be learned and use questions to verify or refute these hypotheses. By asking questions, the information the system deals with can be restricted to what can be used effectively at the time, avoiding the encumbrances of unwieldy searches and large data structures.

The next section reviews strategies for determining what question to pose next. Following that we focus on an economical technique called *conservative selection.* This strategy is exemplified by the MARVIN system (Sammut & Banerji, 1983, 1986) which, when given by the teacher an example of a concept, investigates alternative descriptions for it. Following that a serious problem with straightforward hypothesis-testing procedures is identified: namely that when the concept network is not a strict hierarchy, misleading questions may be formulated. Finally we show how the problem can be solved by detecting when subgoals should be created, and attacking these subgoals by recursively invoking the conservative-selection hypothesis-testing mechanism. The resulting algorithm is implemented in a system called ALVIN, in which a single seed example of a concept, supplied by the teacher, can spawn a whole battery of questions that may provoke a far-reaching investigation of other concepts.

Picking questions to ask

We adopt the standard "generalization as search" paradigm for concept learning (Mitchell, 1982). The object is to locate a concept description within a generalization lattice of alternative descriptions, on the basis of examples classified as positive or negative by a teacher. Normally the teacher also selects the examples, but we allow the learning system to choose them itself and present them for classification.

Within this framework, there are four basic strategies for selecting examples: *randomly*, by *halving*, by *factoring*, and by *conservative selection.*

Choosing examples at random is certainly the simplest, but not usually the most helpful, strategy. Indeed, Van Lehn (1983) has formalized the notion of a sympathetic teacher in terms of "felicity conditions", constraints imposed on or satisfied by a teacher that make learning better than from random examples. However, it is worth noting that any better method, and the whole idea of felicity conditions, implicitly assumes that teacher and student have something in common in terms of how they represent the problem — or at least, that the teacher can see into the student's mind and determine how best to select examples to help him home in on the desired concept. In order to teach people

effectively, one must know something about what they bring to the problem in terms of background knowledge and experience. Equally, to satisfy felicity conditions one must make assumptions about the operation of the learner. This is a good argument for having the system select examples itself.

Suppose a learner contains an explicit representation of the set of descriptions from which the desired concept must be chosen, the so-called "version space" for the concept. Each new example causes the version space to shrink (or remain the same). An efficient strategy, in terms of the number of examples required, is to use each one to cut the version space in half. In other words, one should pick the description whose refutation or verification results in a half-sized version space (Genesereth & Nilsson, 1987). If descriptions are equally likely *a priori*, this strategy will minimize the number of examples required before the desired concept description is identified uniquely. More generally, if the *a priori* distribution is known, one should select each example to maximize the expected information contributed by its classification.

One objection to this strategy is the computational burden of identifying examples which maximize the expected information. In the worst case, every conceivable example must be evaluated with respect to every possible concept description. A second objection is that the examples chosen bear no relationship to each other; the sequence seems to the teacher to be disjointed, unmotivated. For example, Shapiro (1983) used the same idea of binary chopping to narrow down the location of bugs in logic programs; it has been observed (eg by Lloyd, 1986) that this leads to the user being posed a series of questions that seem to be entirely unrelated. This can be disconcerting, and certainly does not resemble the normal problem-solving behavior of people.

The third approach, factoring, breaks the version space into sub-version spaces by independent attributes and applies halving to each of the sub-spaces (Subramanian & Feigenbaum, 1986). This is much more efficient than the general halving method, but still suffers from the drawback of disjointedness. Furthermore, to reap the full benefit examples must be classified by attribute rather than as an overall yes/no decision. This is often infeasible. Moreover, the method is only possible when the concept to be learned can be factored into independent parts. Most problems cannot be decomposed in this way.

The final approach, conservative selection, uses local, directed search to explore the generalization space. Starting with a positive seed example, it climbs one level of the generalization lattice, selects that concept as its current hypothesis, and generates an instance that tests this hypothesis. In effect this follows the "near miss" strategy popularized by Winston (1975) by selecting a new example which differs in just one way from the seed. However, it is stronger in that the new example is guaranteed to differ by a minimal amount. Depending on the teacher's classification of this example, the system continues to generalize or seeks to specialize the current hypothesis. Thus it follows a path through the generalization lattice, step by step, until it arrives at the target concept description. The next step at each stage is guided by the teacher's response to the previously-generated example.

The method of conservative selection is explained in more detail below. Although it generally requires more examples than the halving or factoring approaches (grows linearly rather than logarithmically in the number of possible hypotheses), its operation is more perspicuous. It generates a sequence of examples that indicate to the teacher where it is heading. It avoids large inductive leaps by making steady progress at a local level. It does not rely on special properties of the generalization space (as factoring does). Finally, it is computationally feasible (as opposed to halving and factoring) since generating the next hypothesis is a constant-time operation.

The use of conservative selection

Given a positive example to use as a starting point, the method of conservative selection creates a sequence of examples that lead it towards a unique description for the concept. It accomplishes this by formulating a series of hypotheses about the target description, and finding examples to test each one. We describe each half of this generate-and-test methodology in turn.

Generating candidate hypotheses. At any time, the system has a description called the *currently accepted description.* This is modified little by little until it converges on the target concept. Generalization and specialization are the two means by which the currently accepted description can be modified. To begin, it is initialized to be the seed example.

All one-step generalizations of the currently accepted description are explored in turn. If no more are left, that description itself must be the target concept. Otherwise, the next generalization is selected — call this the *current hypothesis* — to see whether it can replace the current description. It is tested by generating examples and presenting them to the teacher (see next subsection). If it passes, it replaces the currently accepted description and generalization continues. Otherwise alternative specializations of the current hypothesis are explored in turn in case it is a simple overgeneralization. A specialization of a hypothesis is just a restriction of the values of the attributes in the concept description, and can be created by adding an attribute-constraining relation to the hypothesized concept description†. Each specialization is tested by generating and presenting examples. If any one passes the test, the specialization replaces the currently accepted description and generalization continues. If none do, the current hypothesis is abandoned and another is selected. If none remain, the currently accepted description must be the target.

For example, suppose the concepts of *bird, flightless bird* and *aerial bird* depicted in Figure 2a have already been learned. To teach that *birds* are examples of *animals*, the teacher might present *penguin* as an example. The hypothesis generated is the next concept higher up the hierarchy, namely *flightless bird.* When this is verified, it will climb up once more and choose *bird* as the next hypothesis to test. Once that is verified, no more climbing can be done and so *bird* is identified as the target concept.

Testing hypotheses. When a new hypothesis is entertained, it can be tested either *directly* or *indirectly.* For the former, the teacher is asked about the hypothesis itself. For example, suppose *swallow* is presented and the hypothesis becomes *bird.* Then a direct question is "Are all *birds* members of the target concept?" Indirect questioning involves presenting an example of the concept; for example "Is a *robin* an example of the target concept?" Both types are useful. If the teacher is prepared to answer direct questions, the outcome is more reliable because no inference is involved. Moreover, formulating such questions is easy: it merely involves massaging the internal representation of the hypothesis to make it palatable for the teacher.

Generating indirect questions is harder. It is necessary to take the current hypothesis and produce an object which satisfies the following conditions:

† Conservative selection can handle non-factorizable version spaces because of its ability to produce specializations involving non-independent attributes.

Condition 1 It must be covered by the hypothesis (and thus be an example of it);
Condition 2 It must not be covered by the currently accepted description.

For example, suppose that the currently accepted description is *flightless birds* and the hypothesis is *bird* as above. Then animals like *bat*, *trout* and *whale* would fail the first criterion since they are not *birds*, while *chicken* would fail the second since it is a *flightless bird*. An object which satisfies both criteria, for example *kingfisher*, is called a *crucial object*.

Once found, a crucial object is presented to the teacher to ascertain whether it belongs to the target concept. If it does, the current hypothesis becomes the currently accepted description, and generalization continues. If not, attempts are made to specialize the current hypothesis.

Synthesizing crucial objects can be expensive. If the generalization structure is a lattice rather than a tree, some complications emerge which are taken up in the next two sections. Also, the acceptance by the teacher of a crucial object does not guarantee correctness of the current hypothesis, for it may be that the object accidentally satisfies an as-yet-untaught concept. Consequently the use of direct questioning is surer, but more demanding of the teacher since it involves communicating a node of the learner's internal generalization language.

MARVIN: an implementation of conservative selection. An early concept learning system which addressed the problem of formulating questions is MARVIN (Sammut & Banerji, 1983, 1986). This system has several interesting features. First, it represents concepts using Horn-clause logic. Although originally implemented in Pascal, it has been reconstructed in Prolog, a logic programming language based on Horn clauses (Krawchuk, 1987). This version creates actual Prolog clauses and executes them later in the learning process to classify examples. In this sense, it performs inductive program synthesis.

Second, MARVIN immediately adds the concepts it learns to its representation language. Hence previously-learned concepts can be used in the descriptions of new ones. The idea of having generalization languages that grow with time — allowing learning to be sustained — was perhaps the main impetus behind the research.

Third, MARVIN can learn concepts comprising several disjuncts, which, following standard Prolog terminology, we call "clauses". Each is learned independently; by naming them the same the teacher can build up a complex disjunctive concept clause by clause. As each clause is completed it immediately becomes part of the description language. This simple trick makes it possible to learn recursive concepts by teaching the base case first as one clause; it will then be evaluated for inclusion in the description of new clauses just like any other concept. The system is capable of learning recursive procedures for such problems as sorting and list manipulation.

Finally, and most germane to our present topic, MARVIN uses conservative selection to generate examples. To create a crucial object it executes the relevant Prolog concept to obtain an object covered by it, and performs appropriate tests to ensure that the object is crucial.

Misleading questions: the problem with tangled hierarchies

Playing the "twenty questions" game can be frustrating. Sometimes one's current hypothesis is sustained over several consecutive questions, earning steadily increased confidence, only to be demolished by an unexpected answer. Machine learning systems which ask questions can bark up the wrong tree in just the same way. Unlike people, however, they have no way of getting back on track when this happens.

The problem occurs if the hypothesis-testing mechanism implicitly assumes that concepts form a strict hierarchy when they do not. The definition of crucial object given above makes this assumption (as does MARVIN). It is possible to modify the mechanism to make it work correctly by strengthening the notion of crucial object; we do so in the next section. Another tack is to circumvent the difficulty by insisting on hierarchically-structured concept networks; here we show just how unnatural this makes the teaching process.

Hierarchical structuring means that concepts are arranged as a tree rooted at the most general one, with actual domain objects at the leaves. For example, Figure 2a shows some examples of the general concept *bird*, classified into *flightless* and *aerial* birds. This structure quickly becomes untenable when we add other attributes. For example, to include the idea of *swimming birds* while retaining hierarchy requires that the classes *swimming flightless birds* and *swimming aerial birds* be identified (Figure 2b). The non-hierarchical, attribute-oriented, representation in Figure 2c seems much more natural. In practice, concept networks are likely to combine both hierarchical and attribute-like structuring (eg Figure 3).

Practical incarnations of this hypothesis-testing mechanism have assumed a benevolent teacher who is capable of imparting concepts in an order which maintains the tree structure. To do so requires advance planning of the concepts to be taught, and the order in which they will be taught. For example, it is impossible to add the *swimming bird* category to the concept structure in Figure 2a. This places a very considerable burden on the teacher, who might be forgiven for complaining that the task is more like programming than teaching!

What goes wrong if we use the hypothesis-testing method, as outlined above, in non-hierarchical domains? To illustrate the problem, suppose the object *trout* in Figure 3 has been presented and the current hypothesis is *fish*. Under the rules presented so far, *flying fish* might be chosen as an object to test. If it is verified, we would assume that it belongs to the target concept by virtue of being a *fish*. However, the teacher might have verified it not because it is a *fish*, but because it is a *flyer*.

Asking better questions

In order to improve the method it is necessary to suppress misleading questions by trying to avoid presenting objects which support multiple interpretations. To do this, another condition is added to those which characterize crucial objects:

Condition 3 If the object satisfies the target concept, it can do so only because it belongs to the current hypothesis.

For example, again suppose the object *trout* in Figure 3 has been presented and the current hypothesis is *fish*. Then *cod* is a crucial object, while *flying fish* is not since it could be in the target concept by

virtue of being a *flyer* rather than a *fish* (say the target concept was one that included all *flyers*, and, as an exception, the *trout*). Sometimes no crucial objects exist. We define a *significant* object as one which satisfies the first two conditions but not the third. For example, suppose the currently accepted description is *fish* and the hypothesis is *swimmer*. Then *penguin*, *kingfisher*, *whale* and *man* are all significant, but not crucial, objects.

Most often, the question-asking algorithm proceeds as before with the strengthened notion of crucial object. However, if no crucial object exists, a significant object is chosen. If the teacher denies that it is an example of the intended concept, the hypothesis is refuted and specialization is attempted as before. However, if it is declared to be an example, it is necessary to ascertain whether this is because it belongs to the current hypothesis or a competing one.

For example, if the significant object *penguin* is used as a test (in the case above) and the teacher declares it to be positive, we must determine whether that is because it is a *bird* or *flightless bird* rather than a *swimmer*. To decide this, the conservative-selection algorithm is applied to the subdomain including the *flightless bird* and *bird* hypotheses (but not *swimmer*). Here it is profitable to employ conservative selection rather than factoring or halving since normally all hypotheses in this subdomain can be quickly rejected. To do so, each maximally specific hypothesis above the significant object is tested to see if it is refuted. The conservative selection method produces exactly these hypotheses, and saves searching through all the more general ones as the factoring and halving methods would do.

Learning two-leggedness: an example of ALVIN

Conservative selection with improved hypothesis testing has been implemented in a system called ALVIN which learns concepts by asking questions, and can cope with non-hierarchical domains. Like MARVIN, it represents concepts as Prolog clauses; executes them to classify examples; allows learning to be sustained by augmenting its description language with newly-learned concepts; and can learn concepts comprising several disjuncts, including recursive ones. To convey the flavor of interacting with ALVIN, we consider teaching in the domain of Figure 3 the concept of *two-leggedness* enjoyed by *man*, *bat*, and all *birds*. The desired Prolog representation is

```
two-legged(A) :- bird(A).
two-legged(A) :- eq(A, man).
two-legged(A) :- eq(A, bat).
```

Begin by supplying a positive example of *two-leggedness*, say *man*. This could be a positive example for several reasons:

it is a mammal
it is a swimmer
it is a flyer

or perhaps some combination of these:

it is a mammal that swims
it is a mammal that swims and flys
it is a mammal that flys
it is any animal that swims and flys

or perhaps just by virtue of being a *man*. First, ALVIN hypothesizes that all *mammals* are *two-legged*. To test this it must ask about *whale*, *flying squirrel* or *bat*. Although all are significant objects, none is a crucial object. For example, if *whale* turns out to be *two-legged*, it might be because only *swimming mammals* are, not because all *mammals* are. Alternatively, it might be because all *swimmers* are *two-legged*.

Suppose *whale* is selected anyway. As it happens, the teacher declares it is not *two-legged*, and it follows that the current hypothesis (*mammal*) is too general. After every negative example ALVIN tries to specialize its current hypothesis; in this case to *swimming mammals*. This is ruled out by *whale* as well (all the teacher's classifications are stored for possible future use). Consequently specialization continues, to *flying and swimming mammals*.

Since *man* is the sole object covered by this hypothesis, it is rejected. ALVIN retains the most specialized version of a hypothesis so that generalization is conservative. For example, in the future a new flying, swimming mammal might be encountered; and because of this policy it will *not* be automatically assumed to be two-legged. (If the teacher wanted it to be, it would have to be taught explicitly.)

The next hypothesis entertained is *flying mammals*. *Flying squirrel* and *bat* are both crucial objects. Suppose the former is chosen and classified by the teacher as negative. The current hypothesis is further specialized to *swimming and flying mammals* which, since *man* is the only example, is rejected. It cannot be specialized any further, meaning that that *two-leggedness* is independent of whether an object is a *mammal*.

Two alternative generalization branches remain. One, *swimmer*, is ruled out by the *whale*, as also is the more specialized *swimming mammal*. *Swimming and flying mammal* is also ruled out as before. However, the *swimming and flying animal* hypothesis cannot be eliminated so easily. In investigating it, *kingfisher* and *flying fish* are significant objects. The former is chosen (arbitrarily), and the teacher declares it to be *two-legged*. But why? Is it because *kingfishers* swim and fly? Or because they are *aerial birds* or simply *birds*? The last two possibilities must be checked before one can conclude that all animals that fly and swim are two-legged.

Do all aerial birds have two legs? *Robin* and *swallow* are not crucial objects *a priori* because they fly. However, since the *flyer* hypothesis was eliminated earlier, they are now crucial objects. Choosing the former, ALVIN finds that it also is *two-legged*. What about all birds? *Ostrich* and *chicken* are crucial objects. The teacher reports that the former is *two-legged*. Therefore *kingfisher* is an example by virtue of being a *bird*, and not necessarily because it swims and flys. This fact is stored as the first disjunct of the concept representation given above.

However, ALVIN does not rest there, but continues to investigate exactly why *man* is a positive example. It still might be that all flying and swimming animals are *two-legged*. Seeking evidence against this hypothesis, the next significant object is *flying fish*. Since the teacher denies that it is *two-legged*, it follows that not all fish are *two-legged*. It also rules out the *flying and swimming animal* hypothesis. However, that hypothesis can be specialized to *flying and swimming mammal* — we have

encountered this one three times before and will see it twice more. In each case it is refuted for the same reason. In general, when ALVIN encounters a hypothesis more than once, it is refuted just as it was on its first appearance. It is not necessary to store previously-seen hypotheses; remembering previously-seen objects is sufficient.

There is one remaining generalization leg to consider: do all animals that fly have two legs? This is eliminated by the *flying squirrel*. The *squirrel* also rules out the specialization to *flying mammals*, and the *flying and swimming mammal* hypothesis is rejected as before. The other possible specialization of *flyer*, namely *flying and swimming animal*, is refuted by the *flying fish*.

Where does this leave us? All that remains is the possibility that *man* is *two-legged* purely by virtue of being a *man*. This constitutes the second disjunct of the concept representation given above. Altogether, it took seven examples to reach this conclusion:

man	positive	(supplied by teacher)
whale	negative	
flying squirrel	negative	
kingfisher	positive	
robin	positive	
ostrich	positive	
flying fish	negative.	

However, ALVIN not only concluded that *man* is *two-legged*, but that all *birds* are too! Next, the teacher will give *bat* as an example. Meanwhile, several positive and negative examples have been accumulated. Using these, no more questions need to be asked to decide that *bat*, like *man*, is a special case of *two-leggedness*, giving the third and final disjunct of the concept.

Conclusion

We have briefly reviewed the process of formulating questions by similarity-based learners, and focussed on one particular method, conservative selection. While it can be applied straightforwardly in hierarchically-structured domains (and Sammut's MARVIN is an example), the issues involved in general partial orders are more complex. We have illustrated the operation of our implementation, ALVIN, on a particular example because this seems more perspicuous than a formal definition.

We conclude by relating these algorithms to the version-space approach (Mitchell, 1982). MARVIN maintains a version space with a single item in its S set (the specialization boundary), which corresponds to the currently accepted generalization. It considers *all* hypotheses which generalize this item (not just those below the G boundary). In effect, the algorithm is a way of exploring the effect of moving the S boundary one level at a time. However, the significant innovation of *naming* new concepts and adding them immediately to the description language provides a great deal of leverage to the basic generalization process. It allows complex concepts to be constructed from simpler ones; a limited capability for representing disjunctive concepts; and the ability to learn recursive constructs.

Although ALVIN is more complex, it can be viewed as a MARVIN-like mechanism that spawns new version spaces when necessary. Although at any given point it may have created several spaces, it only works on the most recently activated one. However, communication is achieved between spaces because all objects seen are remembered globally.

Thus the conservative-selection method for formulating questions can be characterized in terms of a well-defined theoretical basis. As it stands, however, it is very inefficient — even more so than the version space method itself. The right question now is how to combine this hypothesis-testing approach to selecting examples with more sophisticated learning strategies like explanation- and case-based reasoning.

Acknowledgements

This research is supported by the Natural Sciences and Engineering Research Council of Canada. We gratefully knowledge the work of Bruce MacDonald in helping us to improve our presentation, and the stimulating research environment provided by the Calgary Machine Learning Group.

References

DeJong, G. and Mooney, R. (1986) "Explanation-based learning: an alternative view" *Machine Learning, 1* (2) 145-176.

Genesereth, M.R. and Nilsson, N.J. (1987) *Logical foundations of artificial intelligence*. Morgan Kaufmann, Los Altos, CA, Chapter 7.

Gold, E.M. (1967) "Language identification in the limit" *Information and Control, 10,* 447-474.

Krawchuk, B.J. (1987) "A reconstruction of MARVIN" Research Report 87/280/28, Department of Computer Science, University of Calgary, September.

Lloyd, J.W. (1986) "Declarative error diagnosis" Technical Report 86/3, Department of Computer Science, University of Melbourne, Parkville, Victoria 3052, Australia.

Mitchell, T.M. (1982) "Generalization as search" *Artificial Intelligence, 18,* 203-226.

Pazzani, M.J. (1987) "Inducing causal and social theories: a prerequisite for explanation-based learning" *Proc 4th International Workshop on Machine Learning,* 230-241, Irvine, CA, June.

Rissland, E.A. (1988) "The problem of intelligent example selection" *Int J Man-Machine Studies.*

Sammut, C. and Banerji, R. (1983) "Hierarchical memories: an aid to concept learning" *Proc International Machine Learning Workshop,* 74-80, Allerton House, Monticello, IL, June 22-24.

Sammut, C. and Banerji, R. (1986) "Learning concepts by asking questions" in *Machine learning Volume 2,* edited by R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, pp 167-191. Morgan Kaufmann Inc, Los Altos, CA.

Shapiro, E.Y. (1983) *Algorithmic program debugging*. MIT Press, Cambridge, MA.

Subramanian, D. and Feigenbaum, J. (1986) "Factorization in experiment generation" *Proc Fifth National Conference on Artificial Intelligence,* Morgan Kaufmann, Los Altos, CA.

Utgoff, P.E. (1986) "Shift of bias for inductive concept learning" in *Machine learning Volume II,* edited by R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, pp 107-148. Morgan Kaufmann, Los Altos, CA.

Van Lehn, K. (1983) "Felicity conditions for human skill acquisition: validating an AI-based theory" Research Report CIS-21, Xerox PARC, Palo Alto, CA, November.

Winston, P.H. (1975) "Learning structural descriptions from examples" in *The psychology of computer vision,* edited by P.H.Winston. McGraw Hill, New York, NY.

Witten, I.H. and MacDonald, B.A. (1987) "Concept learning: a practical tool for knowledge acquisition" *Proc Avignon 87 — expert systems and their application,* 1535-1555, Avignon, France, May.

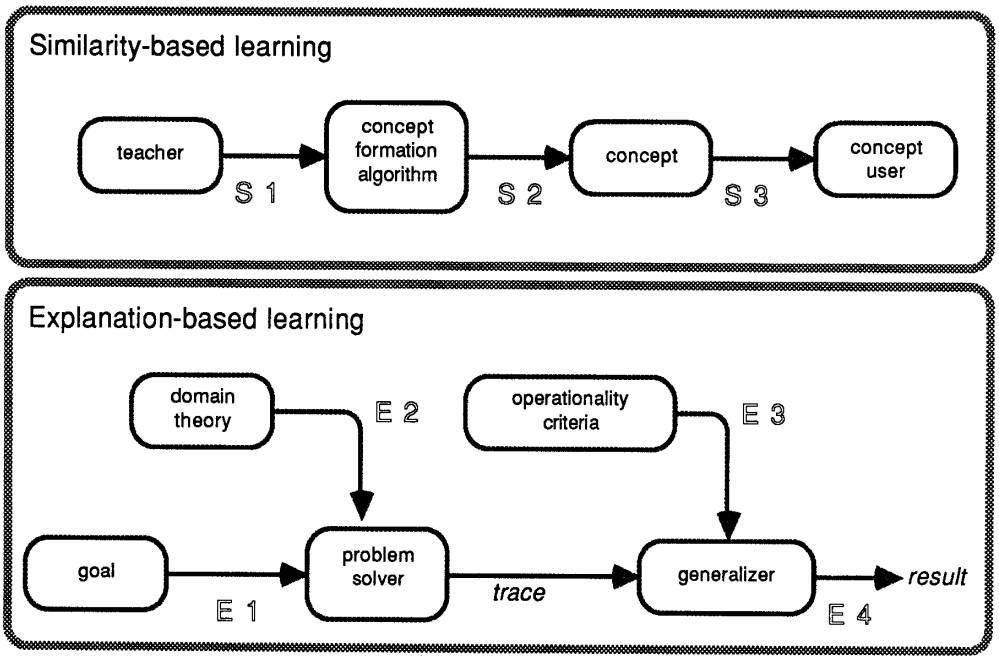

Figure 1 Opportunities for question-asking in similarity- and explanation-based learning

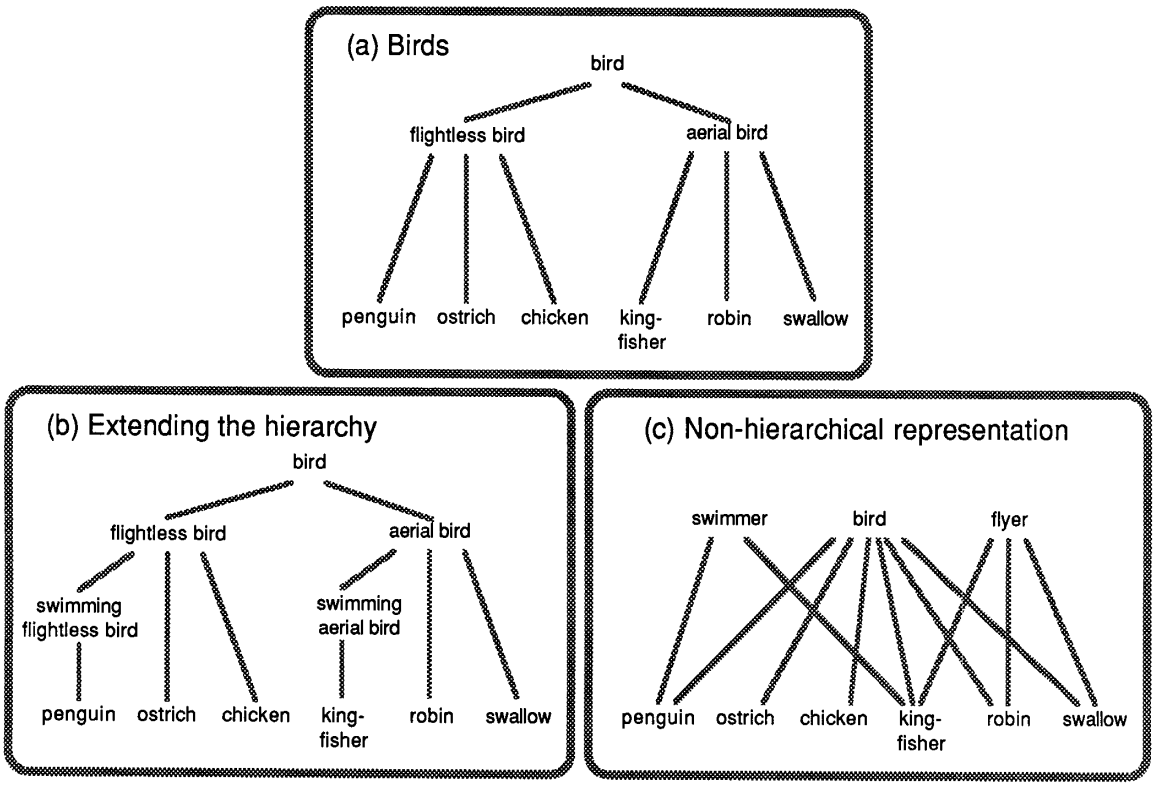

Figure 2 Hierarchical and non-hierarchical domain representations

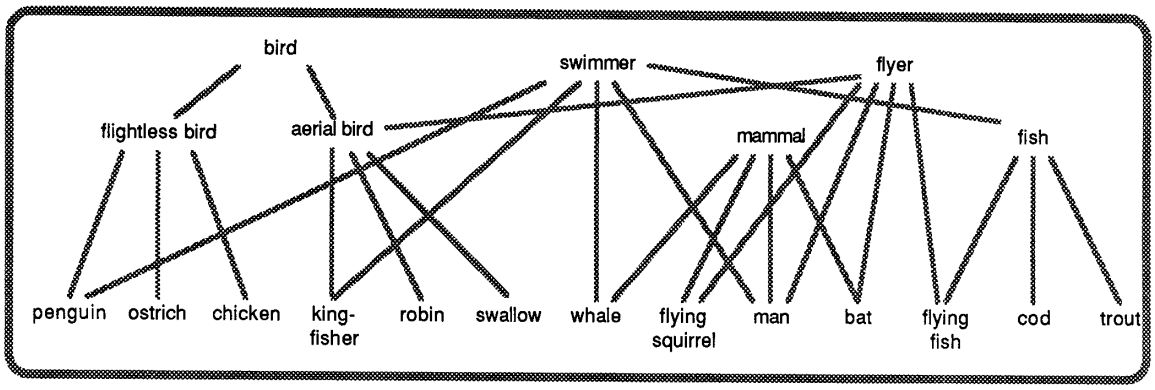

Figure 3 The animal kingdom