Appears in *Proceedings of the Eighth National Conference on Artificial Intelligence*, pp. 815-820, Boston, MA, August 1990..

Changing the Rules: A Comprehensive Approach to Theory Refinement*

Dirk Ourston and **Raymond J. Mooney**
Department of Computer Sciences
University of Texas
Austin, TX 78712
email: dirk@cs.utexas.edu, mooney@cs.utexas.edu

Abstract

This paper presents a comprehensive approach to automatic theory refinement. In contrast to other systems, the approach is capable of modifying a theory which contains multiple faults and faults which occur at intermediate points in the theory. The approach uses explanations to focus the corrections to the theory, with the corrections being supplied by an inductive component. In this way, an attempt is made to preserve the structure of the original theory as much as possible. Because the approach begins with an approximate theory, learning an accurate theory takes fewer examples than a purely inductive system. The approach has application in expert system development, where an initial, approximate theory must be refined. The approach also applies at any point in the expert system lifecycle when the expert system generates incorrect results. The approach has been applied to the domain of molecular biology and shows significantly better results then a purely inductive learner.

Introduction

This paper presents a comprehensive approach to automatic theory refinement. In expert system development, theory refinement occurs when an initial, approximately correct, knowledge base must be refined into a high performance system. The initial knowledge base may correspond to textbook knowledge or rough knowledge from an expert. The refinement process uses a set of training cases to improve the empirical adequacy of the knowledge base, i.e. its ability to reach correct conclusions within its problem space [Ginsberg *et al.*, 1988]. Theory refinement is also required at any point in the lifetime of an expert system when errors are detected in its operation.

Our approach to theory refinement uses a combination of explanation-based and empirical learning methods. Partial explanations of examples and characteristics of the detected errors are used to focus corrections on the failing portion of the theory. Empirical methods are used to learn new rules or modify the premises of existing rules.

The remainder of the paper is organized as follows. The next section presents some background and motivation for our approach. Then we show some simple examples of our system in action, followed by a discussion of the refinement algorithm. Next we present experimental results demonstrating the system's ability to refine a theory for recognizing biological concepts. In the final two sections we present areas for future research and conclusions.

Background

In discussing theories, we will restrict ourselves to propositional Horn clause logic[1], although much of what we say can be applied to other formalisms such as predicate calculus, or even schema representations. We also assume that the inference engine makes a "closed world assumption," i.e. any example provable by the current theory is positive, else it is negative.

Errors in a theory can be classified in terms of the type of examples that are provable. One form of incorrectness is over-generality; i.e. negative examples are provable. This can be caused by two types of errors: 1) an incorrect rule is present in the theory, or 2) an existing rule is missing a constraint from its premise. The

*This research was supported by the NASA Ames Research Center under grant NCC 2-629. Equipment used was donated by the Texas Instruments Corporation.

[1] Our actual representation is somewhat more general than propositional logic since an atomic proposition can also be an attribute value pair or a threshold on a numerical attribute.

other form of incorrectness is over-specificity, i.e. positive examples are not provable. This can also be caused by two types of errors: 1) a rule in the theory has an additional incorrect constraint in its premise, or 2) the theory is missing a rule which is necessary in the proof of certain examples. In general, an incorrect theory can have both overly-general and overly-specific aspects. A comprehensive theory refinement system must be able to handle multiple faults of all types.

Some previous theory refinement systems are only capable of generalizing an overly-specific theory [Wilkins, 1988; Danyluk, 1989; Pazzani, 1989; Ali, 1989] while others are only capable of specializing an overly-general theory [Flann and Dietterich, 1989; Mooney and Ourston, 1989]. For example, the IOU system previously developed by the authors [Mooney and Ourston, 1989] adds constraints to an overly-general theory by using an empirical method to find regularities in the unexplained aspects of the examples.

Many systems do not revise the theory itself but instead revise the *operational definition* of the concept [Bergadano and Giordana, 1988; Hirsh, 1989; Ginsberg, 1988; Shavlik and Towell, 1989; Flann and Dietterich, 1989; Mooney and Ourston, 1989]. Still others rely on active experimentation rather than a provided training set to detect and correct errors [Rajamoney and DeJong, 1988]. Finally, most existing theory correction systems assume a single fault is responsible for each failure [Wilkins, 1988; Danyluk, 1989; Pazzani, 1989].

The system we are developing, called EITHER (Explanation-based and Inductive THeory Extension and Revision), is capable of handling any of the failures described above. The approach can correct multiple faults, and uses one or more failing examples (presented in "batch" format) to learn single or multiple corrections to the theory, as appropriate. The corrections can be made to intermediate points in the theory, rather than strictly involving operational predicates. The method uses positive and negative examples and is able to learn disjunctive rules.

EITHER uses the failures of the explanations created by the domain theory to focus a standard inductive system (currently ID3 [Quinlan, 1986]) to supply corrections to the theory. Because it starts with an initial theory, fewer examples are required to obtain an accurate theory compared with a purely inductive system. In addition, the purpose of our system is to extend the knowledge represented by the initial theory, preserving the structure of the original theory as much as possible. This allows the system to provide better explanations for its conclusions by making use of intermediate concepts in the initial theory.

Examples

Before presenting the details of the system operation, we present some simple examples of how EITHER works. The correct domain theory for the examples, that of drinking vessels, is shown below. This theory is derived from the original cup theory postulated by Winston [Winston *et al.*, 1983].

(stable) $\wedge$ **(liftable)** $\wedge$ **(open-vessel)** $\rightarrow$ **(drinking-vessel)**

(has-bottom) $\wedge$ **(flat-bottom)** $\rightarrow$ **(stable)**

(graspable) $\wedge$ **(lightweight)** $\rightarrow$ **(liftable)**

(has-handle) $\rightarrow$ **(graspable)**

(width small) $\wedge$ **(styrofoam)** $\rightarrow$ **(graspable)**

(width small) $\wedge$ **(ceramic)** $\rightarrow$ **(graspable)**

(has-concavity) $\wedge$ **(upward-pointing-concavity)** $\rightarrow$ **(open-vessel)**

Examples 1 through 6 are a set of examples which are consistent with this correct version of the theory.

1. (+ (has-concavity) (has-bottom) (flat-bottom) (lightweight) (upward-pointing-concavity) (color yellow) (width small) (styrofoam))

2. (+ (has-concavity) (has-bottom) (flat-bottom) (lightweight) (upward-pointing-concavity) (has-handle) (color black) (width medium) (styrofoam))

3. (+ (has-concavity) (has-bottom) (flat-bottom) (lightweight) (upward-pointing-concavity) (has-handle) (color blue) (width large) (ceramic))

4. (- (has-bottom) (flat-bottom) (has-concavity) (lightweight) (upward-pointing-concavity) (width small) (color white) (shape cubical))

5. (- (has-bottom) (has-concavity) (flat-bottom) (upward-pointing-concavity) (lightweight) (width medium) (styrofoam) (color copper))

6. (- (has-bottom) (has-concavity) (flat-bottom) (upward-pointing-concavity) (lightweight) (width medium) (styrofoam) (color blue))

The following subsections are examples of approximate theories which EITHER can be asked to correct. For simplicity, each illustrates a single type of error. Nevertheless, EITHER is designed to handle multiple errors of different types. Note that in the following an *assumption* is an assertion which, if assumed about a particular example, would allow the proof of the example to be completed.

Incorrect Theory: Additional Antecedent

In this case, the theory has been given an extraneous antecedent in the premise of the second graspable rule:

(width small) $\wedge$ (styrofoam) $\wedge$ **(color blue)** $\rightarrow$ (graspable).

The effect of this is that example 1 fails, requiring the assumption *(color blue)* in order for it to be provable. EITHER tentatively removes the assumption *(color blue)* from the graspable rule and checks the negative examples. Since no negative example is provable, EITHER returns the corrected theory with the assumption removed.

Incorrect Theory: Missing Rule

In this case, the rule which accounts for objects which have handles being graspable is omitted from the theory:

[**(has-handle)** $\rightarrow$ **(graspable)**].

Examples 2 and 3 fail. Example 2 requires either the assumption *(width small)* or the assumptions *(width small)* $\wedge$ *(ceramic)* for its proof to be completed. Example 3 requires either the assumption *(width small)* or the assumptions *(width small)* $\wedge$ *(styrofoam)* for its proof to be completed. EITHER selects the assumption *(width small)*, since it represents the smallest change to the theory, and removes it from the rule in which it participates, and checks the modified theory against the negative examples.

In this case, example 5 (a negative example) is provable. EITHER removes the antecedent *(graspable)* from the rule for *liftable*, and re-tests the negative examples. Those that are provable (4, 5 and 6) are used (along with 2 and 3) to discover a new rule for graspable. EITHER adds the rule *(has-handle)* $\rightarrow$ *(graspable)* to the theory.

Incorrect Theory: Missing Antecedent

The theory has been modified such that the second graspable rule is missing the antecedent (width small):

[**(width small)**] $\wedge$ (styrofoam) $\rightarrow$ (graspable).

Negative examples 5 and 6 become provable. EITHER returns the erroneous graspable rule as a candidate rule to retract (this is partially due to the fact that other rules used in the proofs of examples 5 and 6 are used in the proofs of all examples). EITHER removes the graspable rule from the theory and checks to see if all of the positive examples are still provable. Since example 1 is not provable, EITHER sends examples 1, 5 and 6 to the inductive learner for the purpose of learning antecedents to add to the graspable rule. EITHER adds the conjunct *(width small)* to the rule and returns the corrected theory.

Theory Refinement Algorithm

The issues to be addressed by a theory refinement algorithm are: determining that there is an error, identifying the incorrect part of the theory, and finding the required correction. This section discusses the approach to theory correction separately for overly-general and overly-specific aspects. The approach which EITHER uses in either case is one-sided: the algorithm for specializing theories is such that no positive examples are eliminated, and the algorithm for generalizing theories is such that no negative examples are admitted. As a result, the corrections discussed below can be sequentially added to obtain a total correction to an arbitrarily incorrect theory.

Generalizing the Theory

For a theory with overly-specific aspects, the ultimate form of the theory correction will be to add rules, to loosen the constraints of existing rules, or both.

Identifying an Error. The problem of identifying that a theory has overly-specific aspects is straightforward: a positive example fails to be proven.

Finding the Location of the Error. The possible proofs of a given goal in a theory can be represented as an and-or tree (or in the more general case an and-or graph), which we will call the *theory tree*. The original theory tree may be partitioned into a set of and-trees: one for each possible combination of or-branches in the original theory, each one representing a separate possible proof. These are traditionally called proof trees.

For each such proof tree, the leaves of the tree may or may not unify with facts corresponding to the particular example given to the system. In the event that they do not, the system will identify the assumptions required for the proof of the given example. Each such proof is called a *partial proof*, as it requires assumptions in order to be completed. As mentioned in the previous section, assumptions are facts which, if true for the example, would allow a proof to be completed. More importantly, from our point of view, assumptions are literals which, if removed from the premises of the rule in which they are used, would generalize the theory in such a way that the proof attempt would succeed. Constructing partial proofs is a form of *abduction* [Charniak and McDermott, 1985]. In order to restrict the assumptions to observables (assertions expressed using operational predicates) we use *most specific abduction* [Stickel, 1988]. The system which we use to generate partial proofs is a modified version of the ABDUCE system, described in [Ng and Mooney, 1989].

For a complex theory, there will be many such partial proofs and associated assumptions for each unprovable example. In order to minimize the changes to the initial theory, we have adopted the Occam's razor heuristic of finding the minimum number of assumptions required to cover all of the failing examples. Stating the problem as a logical expression we have:

$$E_1 \wedge E_2 \wedge ... \wedge E_n$$

where each of the E's represents the statement that a failing positive example has one or more partial proofs, i.e.

$$E_i \equiv P_{i1} \vee P_{i2} \vee ... \vee P_{im}$$

where the P's represent the statement that a given partial proof for the example is satisfied, i.e.

$$P_{jk} \equiv A_{jk1} \wedge A_{jk2} ... \wedge A_{jkp}$$

where the A_{jkl} represents the *lth* assumption used in the *kth* partial proof of the *jth* example. We then find the minimal set of assumptions, A_{jkl} = True, which satisfy this expression.

The missing rule example, expressed in these terms is:

E_2: (width small) $\vee$ ((width small) $\wedge$ (ceramic))
E_3: (width small) $\vee$ ((width small) $\wedge$ (styrofoam))

and the minimum set of assumptions would consist of the assumption *(width small)*.

In our research, we are comparing two methods for finding the minimum cover of assumptions: a version of the greedy covering algorithm [Johnson, 1974], and the branch and bound algorithm. The greedy algorithm is not guaranteed to find the minimal cover, but will come within a logarithmic factor of it and runs in polynomial time. The branch and bound algorithm is guaranteed to find the minimal cover which accounts for all of the examples, but the process may take exponential time.

Modifying the Theory. Once the minimum cover has been found, the next step is to determine how best to modify the theory so as to account for the failed positive examples. This generalization must also not entail any negative examples.

The heart of the theory modification algorithm is as follows. Assumptions are grouped by the rules in which they participate. The assumptions for each rule are tentatively removed from the antecedents of the rule. If no negative examples become provable, the assumptions are permanently removed. If negative examples become proven, one or more new rules are learned with the same consequent as the current rule. The rules are learned inductively so as to discriminate the appropriate positive and negative examples. The positive examples are those who have a partial proof completed by the assumptions. The negative examples are those that are provable when the current rule consequent is removed from the antecedent of its parent rule[2]. In this way, rules are learned which augment the theory strictly to account for the failure in the given rule chain.

For the missing rule example, EITHER removes the assumption *(width small)* from the graspable rule and tests to see if negative examples are provable. Since they are, EITHER removes *(graspable)* from the rule for *(liftable)* and sees which negative examples are proven (examples 4, 5 and 6). These are passed to the inductive learner along with the positive examples which required the original assumption (2 and 3) in order to learn a new rule with the consequent *(graspable)*. The rule *(has-handle)* $\rightarrow$ *(graspable)* is added to the theory.

There are exceptions to the procedure described above. If all of the antecedents of a rule are removed, and no negative examples become provable, then remove the consequent of the rule from its parent rule instead and recurse. This accounts for the situation where $a \wedge b \wedge c \rightarrow d$ and a b and c have all been removed with no inconsistency with respect to the examples. Since the result is the rule $True \rightarrow d$, which will cause d to always be provable, it is appropriate to remove d from its parent rule instead. This is a less drastic change to the theory, since in the case where the theory is a graph, d may have participated in multiple rules, and we are only interested in those which were actually used in the proofs of the examples.

A second exception is when rules are being learned which are used in the proof of a second, higher level rule. If a majority of the antecedents of a given rule are going to have new rules learned for them on average, then learn a new rule for the consequent of the given rule, instead. As a justification for this heuristic, consider the following example:

$$a \wedge b \rightarrow d$$
$$c \wedge e \rightarrow a$$
$$f \wedge g \rightarrow b,$$

and assume that the theory is missing the rule: $h \wedge i \rightarrow d$. Then an example which is a perfect discriminator for the additional d rule will cause a and b to fail (i.e. h and i will be true but c and e and f and g will not be true). But the positive examples can have arbitrary feature values, as long as they are provable. Any combination of a and b may be provable for examples that are provable

[2]The rule which which precedes the given rule in the rule chain used in the partial proof which includes the assumption.

using $h \wedge i \rightarrow d$. Given all possible examples of $h \wedge i \rightarrow d$, a majority of the time we would be learning new rules for a and b and hence we will learn a new rule for d instead. This form of rule learning is also done recursively, since the higher level rule may also participate in the proof of a yet higher level rule, etc.

Specializing the Theory

In the case of a theory with overly-general aspects, the options are to remove rules or add conjuncts to the premises of rules. An overly-general theory manifests itself by having negative examples that are provable. We would like to modify the theory in such a way that the negative examples are not provable, without losing any of the positive examples. In analogy with the previous section, we would like to make the following statement true:

$$\neg E_1 \wedge \neg E_2 ... \neg En$$

i.e. none of the currently provable negative examples $E_1 ... E_n$ are provable where

$$\neg E_i \equiv \neg P_{i1} \wedge \neg P_{i2} ... \wedge P_{im}$$

i.e. an example is not provable when none of its current proofs are satisfied. And

$$\neg P_{jk} \equiv \neg R_{jk1} \vee \neg R_{jk2} ... \vee \neg R_{jkl}$$

where R_{jkl} is the lth rule used in the kth proof of the jth example, i.e. a proof is not complete if at least one of the rules used in the proof is negated. In analogy with most specific abduction, we consider only rules which occur at the leaves of the proof tree for the particular example. Because of the closed world assumption, the negation of a rule is equivalent to removing it from the theory. Therefore each of the $\neg R_{jkl}$ is equivalent to a rule retraction.

As with assumptions, EITHER forms a minimum cover of rule retractions. If this case, the object is to remove all proofs of all of the provable negative examples. Note that in computing retractions, EITHER removes from consideration those rules which do not have any disjuncts in their proof path to the goal since these rules are needed to prove *any* example.

EITHER removes each of the rules in the minimum cover. If all of the positive examples remain provable, then the rule is permanently removed. If any positive examples fail to be proven, then additional antecedents are added to the rule to prevent it from providing proofs for negative examples while still providing proofs for positive examples. An appropriate set of antecedents is found by giving the inductive learner the positive examples which fail be proven with the rule removed and the negative examples which used the rule in a proof. The features used in the original rule are removed from the examples before they are sent to the inductive learner, and then added back in to the rule that is learned. In this way, we are guaranteed that the learned rule, which replaces the original rule in the theory, is a specialization of the original rule.

For the missing antecedent example, EITHER removes the rule *(styrofoam)* $\rightarrow$ *(graspable)* from the theory since this is the only disjunctive rule required in the proofs of the negative examples. Since a positive example becomes unprovable when this is done, EITHER sends the failing positive example and the provable negative examples to the inductive learner after removing the feature *(styrofoam)* from the examples. The inductive learner learns the rule *(width small)* $\rightarrow$ *(graspable)* and EITHER adds the feature *(styrofoam)* back in to form the rule *(width small)* $\wedge$ *(styrofoam)* $\rightarrow$ *(graspable)* which replaces the original rule in the theory.

Experimental Results

The EITHER algorithm was tested on a theory used for recognizing biological concepts in DNA sequences. The original theory is a modified version of the theory described in [Towell *et al.*, 1990]. The goal of the theory is to recognize *promoters* in strings composed of nucleotides (one of A, G, T, or C). A promoter is a genetic region which initiates the first step in the expression of an adjacent gene (*transcription*), by RNA polymerase. We modified the original theory by removing the tests for conformation in order to improve its tractability. The reduced theory then corresponds to "Pribrow Boxes". The input features are 57 sequential DNA nucleotides. The examples used in the tests consisted of 53 positive and 53 negative examples, assembled by a biologist from the biological literature. The initial theory classified four of the positive examples and all of the negative examples correctly. This indicates that the initial theory was entirely overly-specific.

Figure 1 shows the performance results obtained when EITHER was used to refine this theory. In each test, performance was measured against twenty five test examples. The number of training examples was varied from one to 80, with the training and test examples drawn from the entire example population, with no overlap. The results were averaged over 50 samples. The figure shows that using the approximate theory provides a significant performance advantage, and that this advantage is maintained over the entire training interval. An analysis of the runs showed that EITHER was modifying both leaf level and intermediate concepts in obtaining these results.

A one-tailed Student t-test on paired differences showed that the superior performance of EITHER is statistically significant at the 1% level for every point plotted on the learning curves. After 80 training ex-

Appears in *Proceedings of the Eighth National Conference on Artificial Intelligence,* pp. 815-820, Boston, MA, August 1990..

amples, the 95% confidence interval for the difference between EITHER and ID3 is 5.8% to 10.0% (i.e. with a probability of 0.95 EITHER's accuracy is between 5.8 and 10.0 percentage points higher than ID3's).

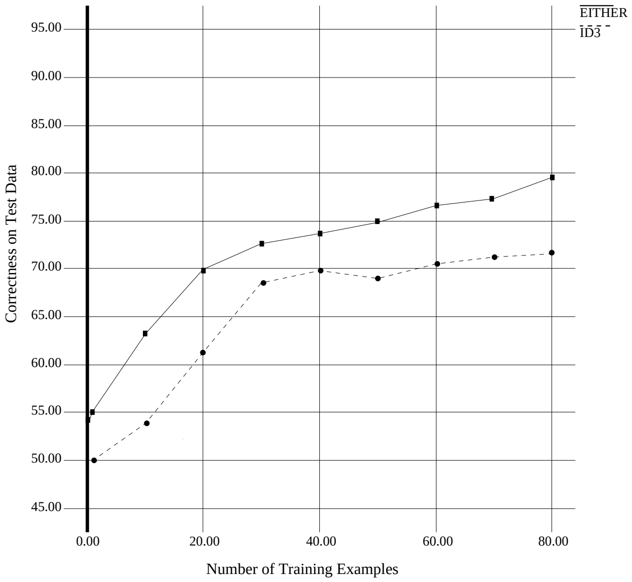

Figure 1: EITHER Results for the DNA Theory

Future Research

Empirical tests on additional domains and theoretical analysis of computational complexity and learnability issues are obvious areas for future research. Other directions include extending the approach to deal with noisy data and predicate calculus and allowing the initial theory to be used as a source of rules for constructive induction.

Conclusions

This report has outlined a technique for theory revision which combines elements of empirical and explanation-based learning. The approach attempts to preserve the structure of the theory as much as possible so that the intermediate concepts represented in the original theory are preserved. Since the technique uses an initial theory it shows definite performance advantages when compared to a purely inductive system. Unlike other theory refinement systems, the proposed approach is capable of handling multiple faults and handles both overly-general and overly-specific aspects of an incorrect theory.

Acknowledgments

Jude Shavlik provided us with the theory and data for the DNA tests, and also helped us to understand the background behind the theory. Michiel Noordewier actually prepared the original theory and data base.

References

[Ali, 1989] Karmal M. Ali. Augmenting domain theory for explanation-based generalization. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 40–42, Ithaca, NY, June 1989.

[Bergadano and Giordana, 1988] Francesco Bergadano and Attilio Giordana. A knowledge intensive approach to concept induction. In *Proceedings of the Fifth International Conference on Machine Learning (ICML-88)*, pages 305–317, Ann Arbor, MI, June 1988.

[Charniak and McDermott, 1985] Eugene Charniak and Drew McDermott. *Introduction to AI.* Addison-Wesley, Reading, MA, 1985.

[Danyluk, 1989] Andrea Pohoreckyj Danyluk. Finding new rules for incomplete theories: Explicit biases for induction with contextual information. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 34–36, Ithaca, NY, June 1989.

[Flann and Dietterich, 1989] Nicholas S. Flann and Thomas G. Dietterich. A study of explanation-based methods for inductive learning. *Machine Learning*, 4(2):187–226, 1989.

[Ginsberg *et al.*, 1988] Allen Ginsberg, Sholom M. Weiss, and Peter Politakis. Automatic knowledge based refinement for classification systems. *Artificial Intelligence*, 35:197–226, 1988.

[Ginsberg, 1988] Allen Ginsberg. Theory revision via prior operationalization. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 590–595, St. Paul, MN, August 1988.

[Hirsh, 1989] H. Hirsh. *Incremental Version-Space Merging: A General Framework for Concept Learning.* PhD thesis, Stanford University, Palo Alto, CA, June 1989.

[Johnson, 1974] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[Mooney and Ourston, 1989] Raymond J. Mooney and Dirk Ourston. Induction over the unexplained: Integrated learning of concepts with both explainable and conventional aspects. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 5–7, Ithaca, NY, June 1989.

[Ng and Mooney, 1989] Hwee Tou Ng and Raymond J. Mooney. Abductive explanations for text understanding: Some problems and solutions. Technical Report AI89-116, Artificial Intelligence Laboratory, University of Texas, Austin, TX, August 1989.

[Pazzani, 1989] Michael J. Pazzani. Detecting and correcting errors of omission after explanation-based learning. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 713–718, Detroit, MI, August 1989.

[Quinlan, 1986] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[Rajamoney and DeJong, 1988] Shankar A. Rajamoney and Gerald F. DeJong. Active explanation reduction: An approach to the multiple explanations problem. In *Proceedings of the Fifth International Conference on Machine Learning (ICML-88)*, pages 242–255, Ann Arbor, MI, June 1988.

[Shavlik and Towell, 1989] Jude W. Shavlik and Geoffrey G. Towell. Combining explanation-based learning and artificial neural networks. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 90–92, Ithaca, NY, June 1989.

[Stickel, 1988] M. E. Stickel. A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. Technical Report Technical Note 451, SRI International, Menlo Park, CA, September 1988.

[Towell *et al.*, 1990] Geoffrey G. Towell, Jude W. Shavlik, and Michiel O. Noordewier. Refinement of approximate domain theories by knowledge-based artificial neural networks. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 861–866, Boston, MA, July 1990.

[Wilkins, 1988] David C. Wilkins. Knowlege base refinement using apprenticeship learning techniques. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 646–651, St. Paul, MN, August 1988.

[Winston *et al.*, 1983] Patrick H. Winston, Thomas O. Binford, Boris Katz, and Michael R. Lowry. Learning physical descriptions from functional definitions, examples, and precedents. In *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83)*, pages 433–439, Washington, D.C., August 1983.