A Methodology for Evaluating Theory Revision Systems: Results with Audrey II*

James Wogulis and **Michael J. Pazzani**
Department of Information & Computer Science
University of California
Irvine, CA 92717
E-mail: wogulis@ics.uci.edu pazzani@ics.uci.edu

Abstract

Theory revision systems are learning systems that have a goal of making small changes to an original theory to account for new data. A measure for the distance between two theories is proposed. This measure corresponds to the minimum number of edit operations at the literal level required to transform one theory into another. By computing the distance between an original theory and a revised theory, the claim that a theory revision system makes few revisions to a theory may be quantitatively evaluated. We present data using both accuracy and the distance metric on AUDREY II, a first-order theory revision system.

1 Introduction

Many recent learning systems are designed to take advantage of a user supplied, approximate concept definition (also called an incomplete or incorrect domain theory) to guide or constrain the learning process. Pazzani and Kibler [1992] state that the reason for providing a domain theory is "to increase the accuracy of learned rules." A variety of systems with different learning mechanisms share this objective, e.g., ML-SMART [Bergadano and Giordana, 1988], A-EBL [Cohen, 1992a], GRENDEL [Cohen, 1992b], IOE [Flann and Dietterich, 1989], IVSM [Hirsh, 1989]. These systems are typically evaluated by comparing the accuracy of the learned rules with and without an initial theory.

Some learning systems, called theory revision systems, impose an additional constraint on the learner. Mooney and Richards [1992] state that the goal of theory revision systems is "to minimally modify the theory to correctly classify all of the examples." The first theory revision systems such as EITHER [Ourston and Mooney, 1990] and KBANN [Towell, 1991] were restricted to revising theories in propositional logic. More recently, several systems such as KR-FOCL [Pazzani and Brunk, 1991], FORTE [Richards and Mooney, 1991; Richards, 1992] and AUDREY [Wogulis, 1991] have been introduced that revise theories expressed as Horn clauses. In spite of the constraint that the learned theory be a minimal revision of the given theory, the only systematic evaluation of these systems has been with respect to the accuracy of the learned theory.

Here, we propose a distance metric between Horn clause theories that may be used to evaluate how well theory revision systems meet their stated goals. The measure is defined as the number of edit operations (i.e., additions, deletions or replacements of literals) that are needed to transform one theory into another. The notion of edit-distance has been studied by others [Tai, 1979; Shasha and Zhang, 1990] and applied to problems in areas such as pattern matching, parsing, and comparing secondary structures of RNA. Without such an evaluation, a system that ignores the initial theory and runs a purely inductive learner on the training data might also qualify as a theory revision system.

In the rest of this paper, we review how Horn clauses are used to represent domain theories, we describe in detail why the minimal modification constraint is useful, and we define a measure of the distance between two theories. This measure is used to define the appropriateness of a revised theory with respect to the initial theory. We then present a theory revision system, AUDREY II, and show its performance on a number of domains with respect to both accuracy and appropriateness of revision. Finally, we compare AUDREY II's performance with other theory revision systems.

Table 1: Domain theory for concepts **grandparent** and **parent**.

```
grandparent(G,C) :- parent(G,P), parent(P,C).
parent(X,Y):- father(X,Y).
parent(X,Y):- mother(X,Y).
```

2 Representation

A common method used by first-order inductive learners such as FOIL [Quinlan, 1990], FOCL [Pazzani and Kibler, 1992], GOLEM [Muggleton and Feng, 1990] and first-order theory revisers such as KR-FOCL, FORTE and AUDREY is to represent domain theories as a set of first-order Horn clauses and instances as ground literals.

*This research was supported in part by grant number F49620-92-J-0430 from AFOSR.

A single concept is represented as the set of Horn clauses in the domain theory that have the same predicate symbol (and arity) for the head of the clause. For example, the concept **grandfather** would be represented by the set containing the first of the three clauses in the domain theory in Table 1 and the concept **parent** would be represented by the last two.

An incorrect domain theory is one that classifies a negative example as a positive example. An incomplete domain theory is one that classifies some positive instances as negative. Note that a domain theory may be both incomplete and incorrect.

3 Theory Revision

Theory revision systems are given an initial domain theory that is incorrect and/or incomplete, and a set of positive and negative training examples and produce a revised theory. In this paper, we propose that an appropriately revised theory is one that is closest to the initial theory and one that makes the fewest errors on unseen examples drawn from the same distribution as the training examples.

Using distance as one factor to measure the quality of a revised theory favors theories that preserve as much of the initial theory as possible. There are several reasons that this is desirable:

- Such revisions are more likely to be understood by experts who presumably understand the initial theory, because they represent minor variants of the experts' theories. An expert may be unwilling to accept a drastic change to a theory, especially if adding or removing a few training examples results in yet another drastically different theory.
- The abstractions from the initial theory may be retained in the revised theory. Many learning systems (e.g., FOCL and A-EBL) that use a domain theory to guide learning express the learned concept in terms of the operational predicates. This results in a "flat" theory, expressed solely in terms of "observables" without intermediate conclusions. In contrast, a theory revision system should use the abstract predicates from the initial theory, provided they are useful in making classifications.
- Abstractions from the initial theory that are retained in the revised theory can support the generation of meaningful natural language explanations of the system's translations [Swartout, 1981].

We note that using distance favors theories that make small incremental changes rather than major re-writes of the initial theory. While we feel this is desirable in general, there are situations, perhaps corresponding to paradigm shifts [Kuhn, 1962], in which an entire theory must be discarded and replaced by a drastically different theory. We do not consider this issue in detail, but note that it might be addressed by introducing a dimension, such as elegance or simplicity, that may be combined with accuracy and distance to evaluate revised theories.

One can use the distance metric to compare different revisions of the same initial theory to determine which revision is closest to the initial theory. When used together with accuracy, distance provides a better overall measure of the quality of the revised theories. Clearly, a theory that performs better on both accuracy and distance is to be preferred. If theory A is more accurate than theory B but theory B is closer to the initial theory, then the issue becomes murkier. Since these measures are orthogonal, preferring one theory over another depends on the user's trade-off between the two goals.

4 Theory Distance Metric

Our goal is to define a metric analogous to accuracy that can be used to measure the quality of revisions made by theory revision systems. Whereas the accuracy of a revised theory measures its semantic correctness, we define a measure of the syntactic closeness between the two theories as the distance between the initial and revised theories. The distance between two theories is defined as the minimum number of revisions required to transform one theory into another.

4.1 The distance between two theories

As defined above, a domain theory is a set of Horn clauses which is partitioned into concepts. A concept $K_{pred}(T)$ for theory T is defined as the set of clauses in the theory T with the same predicate symbol $pred$ and arity for the head of the clause: $K_{pred}(T) = \{C \in T \mid C \equiv pred(\cdots) \leftarrow L_1, \cdots L_n.\}$. For example, the clauses `append([],[],[])` and `sort([],[])` belong to different concepts. We define the distance between two theories to be the sum of the distances between corresponding concepts from each theory.

$$dist(T_1, T_2) = \sum_{pred \in T_1 \cup T_2} dist(K_{pred}(T_1), K_{pred}(T_2))$$

This definition of distance does not compare clauses from different concepts so there is no defined distance between the clauses for `append` and `sort` given above.

Since a concept is a set of clauses, there may be many ways in which one concept could be transformed into another. When measuring the distance between concept A and concept B, there are two main cases to consider. A clause C_A from A could be the source of many (including zero) clauses in B: $C1_B, ...Cn_B$, or similarly many clauses in A ($C1_A ...Cn_A$) could be the source of a single clause C_B in B. We define the distance between two concepts to be the minimum sum of clause distances for all mappings between clauses of each concept.

$$dist(A, B) = \min_{map \in mappings(A,B)} \left[\sum_{\langle C_A, C_B \rangle \in map} dist(C_A, C_B) \right]$$

Since there is no way to know which clauses in one theory should correspond to which clauses in another, we define the distance between two concepts to be the minimum sum of distances for all possible mappings between clauses from each concept. Clauses may be mapped to the empty clause which corresponds to clause deletion.

The set of mappings between clause sets A and B can be computed as follows. First, form the cartesian product of clauses from A and B. For example if $A = \{a_1, a_2\}$

and $B = \{b_1, b_2\}$ then one element of $A \times B$ would be $\langle a_1, b_2 \rangle$. Next, compute the power set of $A \times B$. A single mapping is formed from one element of this power set, e.g., $map = \{\langle a_1, b_1 \rangle, \langle a_2, b_1 \rangle\}$. For any $a_i \in A$ or $b_j \in B$ not appearing in map, add the elements $\langle a_i, \emptyset \rangle$ and $\langle \emptyset, b_j \rangle$ to map where $\emptyset$ represents the empty clause. Therefore, we would transform map above into $map = \{\langle a_1, b_1 \rangle, \langle a_2, b_1 \rangle, \langle \emptyset, b_2 \rangle\}$. This mapping is interpreted to mean B can be revised into A by transforming clause b_1 into clauses a_1 and a_2 and by deleting clause b_2. In general, the number of mappings between clause sets A and B is $2^{|A||B|}$.

For our present purposes this definition will be at the literal level: i.e., how many additions, deletions or replacements of literals are needed to transform one clause into another.[1] However, differences in the order of literals in a clause and renaming of logical variables are ignored. We define the $dist(C_1, C_2)$ to be the minimum number of additions, deletions or replacements of literals needed to transform clause C_1 into clause C_2 modulo variable names and literal ordering in the bodies of the clauses. For example, the distance between the two following two clauses is 1:

```
c1: grandfather(X,Y) :- sister(X,Y),
                        father(X,Z).
c2: grandfather(A,B) :- father(A,C),
                        parent(C,B).
```

Without regard to the variable names or goal ordering, clause *c1* can be transformed into *c2* by replacing `sister(X,Y)` with `parent(Z,Y)`.

It is important to note that our metric for measuring the distance between two theories is used during evaluation and not during the learning process. As we have implemented it, the metric is exponential and somewhat costly to compute but not prohibitive on the domains we have worked with. We suspect the complexity of our problem to be polynomial since it is closely related to the assignment problem [Lawler, 1976]. We are in the process of determining the complexity of this problem.

4.2 Relation Between Distance and Accuracy

It is important to note that the definitions of accuracy and distance are orthogonal to one another and thus complementary. That is to say that two theories may have the same accuracy, even making the same classification of every example, and yet there may be a large syntactic difference between the two theories. An example of this is the difference between theories 1 and 2 in Table 2, both of which represent a correct theory for positive, even integers.[2]

Similarly, two theories may be close to one another syntactically but have entirely different semantic meaning. An example of this is theories 1 and 3 in Table 2. Both theories are quite close syntactically but have very different meaning and hence accuracy.

Table 2: Three theories for positive, even integers.

```
Theory 1:
    poseven(X) :- greater(X,0), even(X).
Theory 2:
    poseven(X) :- modulo(X,2,Y), equal(Y,0),
                  not(greater(Y,X)).
Theory 3:
    poseven(X) :- less(X,0), even(X).
```

4.3 Uses of the Distance Metric

Our primary use of the distance metric will be to compare revisions of the same initial theory by theory revision systems to determine which revision is closest to the initial theory. This will allow one to evaluate the claim that theory revision system A produces better revisions than theory revision system B.

However, if the correct theory is known (e.g., if the incorrect theory were formed for evaluation purposes by introducing errors into the correct theory), then there are two additional uses of the distance metric.

First, the distance between the initial theory and the correct one provides a measure of the syntactical corruptness of the initial theory. We believe that this will provide a more useful measure of the difficulty of a theory revision problem than just the accuracy of the initial theory. Presumably, the more corrupt a theory is, the harder it will be to revise.

Second, the distance between the revised theory and the correct one can be used in conjunction with accuracy to measure how good a learning system is at replicating human created theories. Currently, knowledge engineers perform the theory revision task manually. A prototype expert system is often built, and then it is refined to perform better on a set of test cases. The initial theory and test cases can serve as input to a theory revision system and the automatically derived theory can be compared to the theory produced manually. The distance between the automatically and manually revised theories measures the system's ability to find theories similar to ones built by experts.

4.4 Some Drawbacks

Theory revision systems require evaluation beyond simply measuring the accuracy of the theories produced. A measure of the type and quality of revisions made is also needed. Although the distance metric proposed here is useful toward that end, it is not a panacea. Since distance is a syntactic measure, it does not handle some theories one would like to consider equivalent. For example, the learner may have clauses containing literals with equivalent meaning such as `less(X,Y)` and `geq(Y,X)`. Comparing two theories at the syntactic level will not consider these two literals equivalent. We might address

[1] Another possible definition could be more detailed and include the number of changes to terms within a literal. We will ignore this complication since the theories used by theory revision systems so far do not include complex literals such as `pred(f(A),[A|B])`. We note that these might be handled within this framework by breaking a complex literal into several simpler ones [Norvig, 1992] `pred(V1,V2) & V1 = f(A) & V2 = [A|B]`.

[2] $modulo(A, B, M)$ is defined such that M is the remainder of A divided by B.

this in the future by providing additional equality axioms to determine the equivalence between two literals.

5 The Audrey II System

In this section we present AUDREY II which performs first-order theory revision.[3] The system operates in two main phases. First, the initial domain theory is specialized to exclude covering any negative examples. Next, the revised theory is generalized to cover all of the positive examples without covering any negatives.

AUDREY II uses four operators during theory revision: delete a clause, add a literal, delete a literal, and add a clause. A blame assignment mechanism guides the application of these operators and revisions are chosen based on their improvement to the theory's accuracy.

5.1 Specialization Phase

The specialization phase uses a hill-climbing approach to specialize the theory until no negative examples are covered. At each step a best clause to specialize is chosen, the clause is specialized and the process repeats. The best clause is the one that contributes to the most number of negative examples being incorrectly classified and is required by the fewest number of positive examples.

If the clause is not needed by any of the positive examples, then it can be safely deleted. Otherwise, it is specialized using a FOIL-like operator that adds literals to the clause one at a time until none of the negatives the clause was responsible for misclassifying are covered. This could lead to the introduction of several new clauses that are more specific than the original. For example, the overly general clause: `uncle(U,N) :- male(U)` might be specialized into the two clauses:

```
uncle(U,N) :- male(U),
              sibling(U,P), parent(P,N).
uncle(U,N) :- male(U), married(U,M),
              sibling(M,P), parent(P,N).
```

5.2 Generalization Phase

During the generalization phase, an uncovered positive example is randomly chosen and the theory is generalized to cover the example. The process repeats on any remaining uncovered positives until all are covered. To cover the example, AUDREY II first applies the abductive process used in AUDREY [Wogulis, 1991] to find a single literal to assume that, if true, would allow the example to be explained. The assumption identifies where the domain theory should be repaired and AUDREY II extends the theory so that the example can be correctly classified without relying on the abductive mechanism.

The assumed literal is deleted from the clauses in which it was used if the resulting theory does not decrease coverage of the positive examples. If deleting the literal from a clause decreases the coverage of the positive examples, then AUDREY II tries replacing the literal with a new conjunction of literals determined by FOIL. For example, the incorrect clause `uncle(U,N) :- male(U), female(N)` might be transformed into the following two clauses by replacing the literal `female(N)` with new literals:

```
uncle(U,N) :- male(U),
              sibling(U,P), parent(P,N).
uncle(U,N) :- male(U), married(U,M),
              sibling(M,P), parent(P,N).
```

If deleting or replacing the assumed literal fails to improve coverage, then AUDREY II uses its FOIL-like component to learn a new set of clauses for proving the literal. For example, a theory only containing the clause `grandfather(A,B) :- father(A,C), parent(C,B)` is too specific if it contains no clauses for `parent`. Therefore, positive examples of `grandfather` would require making an assumption about the `parent` relationship and could lead to learning new clauses for `parent`:

```
parent(P,C) :- father(P,C).
parent(P,C) :- mother(P,C).
```

6 Experimental Evaluation

In this section, we present some experimental results of the AUDREY II system using accuracy and theory distance as a performance measure. In the first experiment we compare AUDREY II and FOCL on Pazzani and Brunk's [1991] student-loan domain measuring accuracy, distance of revised theory from original and distance from the correct theory each as a function of the number of examples. In the second experiment we compare the results of AUDREY II and FORTE on four different, incorrect domain theories for the King-Rook-King problem. To conserve space, we do not show the original theories here; they can be found in Pazzani and Brunk [1991] and Richards [1992].

6.1 Student Loan Domain

Pazzani and Brunk [1991] presented a domain theory for determining if a student needed to make payments on a student loan. The theory contained sixteen clauses and was up to four levels deep. Four different types of errors were introduced into the theory: one clause was missing a literal, a whole clause was missing, one clause had an extra literal and one concept had an extra clause. We compare AUDREY II to FOCL to emphasize the difference between revising a theory and using a theory to guide induction.

The experiment we ran consisted of giving AUDREY II and FOCL the initial incorrect theory and examples chosen from a pool of 50 students. For each different number of input examples, we measured the revised theories in terms of accuracy on unseen examples, distance from the original theory and distance from the correct theory. The results are shown in Figures 1, 2 and 3 and are averaged over 20 trials.

Figure 1 shows that AUDREY II performs better on accuracy than FOCL up to 45 examples at which point both systems achieve 100% accuracy. In Figures 2 and 3, we can see that the distances from both the original and correct theories for FOCL are much greater than for

[3] Audrey II differs from Audrey [Wogulis, 1991] primarily in its use of a Foil-like component for inductive learning rather than forming least-general generalizations. Also, Audrey II can specialize clauses by adding literals which Audrey wasn't able to do.

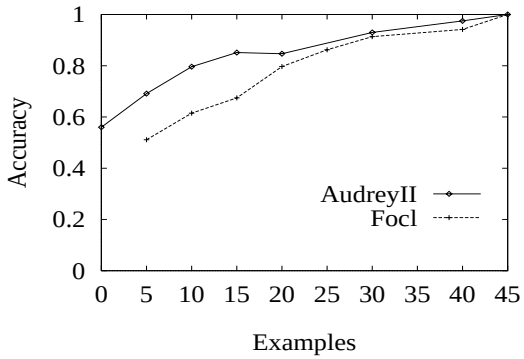

Figure 1: Accuracy for AUDREY II and FOCL on student loan domain.

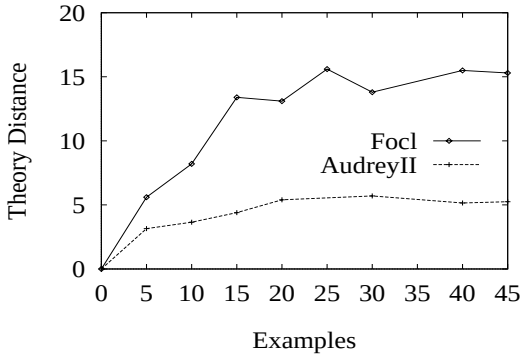

Figure 2: Distance between original and revised theory for student loan domain.

AUDREY II. If we look at the distance of the revised theory from the original as a function of the number examples (Figure 2), we see it rising for AUDREY II from 0 to around 25 examples after which it levels off. We stated earlier that a "good" revised theory should be accurate and be close to the original theory. However, we see that early on the theory isn't very accurate but the distance to the original theory is small. When the theory is 100% accurate (after 45 examples), we see that its distance from the original has increased. In fact, according to our distance metric, the correct, 100% accurate theory given by Pazzani and Brunk [1991] should have a distance of four from the original theory, while the 100% correct theory learned by AUDREY II has an average distance of slightly more than five.

The results comparing the accuracy of FOCL and AUDREY II indicate another benefit of the bias towards minimal revisions of theories. AUDREY II does not delete any of the domain theory unless doing so improves accuracy on the training set. In contrast, FOCL learns rules that discriminate positive from negative examples. When there are few training examples, it ignores those portions of the domain theory that are not needed to explain any of the positive examples in the training set.

Figure 3 shows the distance between the revised and

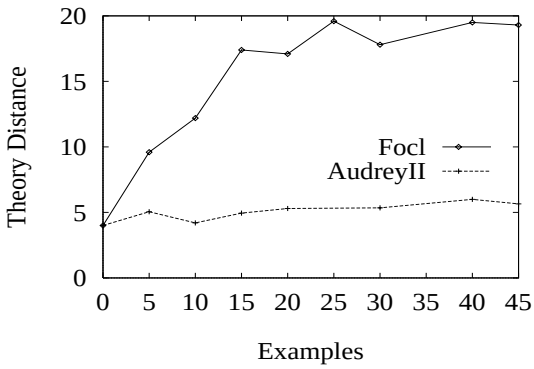

Figure 3: Distance between correct and revised theory for student loan domain.

correct theory. We notice that the original theory has a distance of four from the correct theory. As learning proceeds, the distance between the revised and correct theory actually increases. One would hope that this distance would, in fact, go to zero as accuracy approaches 100%. Unfortunately, for this problem, AUDREY II's distance from the correct theory is approximately 5.5 when it is 100% accurate. In this case, the theory revision problem is under-constrained. For example, the correct theory should contain the following clauses:

```
continuously_enrolled(S) :-
        never_left_school(S),
        part_time_student(S).

part_time_student(S) :-
        enrolled(S,School,U),
        school(School), U > 5.
```

While AUDREY II typically learns

```
continuously_enrolled(S) :-
        part_time_student(S).

part_time_student(S) :-
        enrolled(S,School,U),
        school(School), U > 5,
        never_left_school(S).
```

Although these two definitions are equivalent with respect to accuracy, the latter definition does not reflect a proper definition of part-time student that an expert would be willing to accept. It is exactly this type of difference that the distance metric proposed here is intended to capture. The problem arises in this domain because there are two places where one could add a literal to achieve the same accuracy. While both are equally distant from the original theory, one makes the revised theory closer to the known correct theory, while the other moves it further away. This problem would not occur if the syntactic differences between the theories resulted in differences in accuracy. This could be accomplished in AUDREY II by providing examples for intermediate concepts (e.g., examples of part-time students), or if the predicate `part_time_student` were used elsewhere in the theory.

6.2 King-Rook-King Domain

The king-rook-king problem [Muggleton and Feng, 1990; Pazzani and Kibler, 1992; Richards, 1992] requires the learner to recognize illegal chess board positions containing only a white king and rook and a black king. An illegal position is one in which the black king is in check or where two pieces occupy the same square. Richards used this domain to test FORTE and provided sample revised theories for four different corrupt versions of the correct theory. In this section, we review the FORTE system and compare the performance of AUDREY II and FORTE on the four[4] KRK domains described in Richards [1992].

6.2.1 FORTE

FORTE is a first-order theory revision system that uses hill climbing and a larger set of more complex revision operators. Revision points in the theory are found by recording the succeeding clauses in the proof of a negative example and the failing points in the failed proof of a positive example. FORTE determines all the revision points for all positive and negative examples and applies the various revision operators to each point. The revision with the best improvement in theory accuracy and simplicity is retained and the hill climbing proceeds. FORTE uses a large set of operators: delete a clause, delete a literal, delete multiple literals, add a literal, add a clause, identification, absorption [Muggleton and Buntine, 1988], multiple literal deletion, and some that combine several operators into more complex ones.

There are a couple of major differences between AUDREY II and FORTE. First, AUDREY II uses an abductive process that more accurately identifies where the theory should be modified than does FORTE's revision points. Second, FORTE has a larger number of more complex operators, some of which may make major structural changes to the theory such as identification and absorption. Finally, FORTE was evaluated on the accuracy and size of the theories it produced and not on the closeness to the original theory or degree of revision.

6.2.2 Comparison

We would like to compare FORTE and AUDREY II as theory revision systems which involves comparing the two on accuracy and theory distance. Unfortunately, as with most theory revision systems, the published results on FORTE don't include the distance between the original and revised theories. However, Richards [1992] did provide revised theories for each of four corrupt king-rook-king domains. The results reported gave the initial and final theories for each of four corrupt theories after seeing 50 of 2000 randomly chosen instances. The results of only one trial for each theory were reported. To compare the two systems, we gave AUDREY II 50 randomly chosen examples for each of the four corrupt theories and averaged the results over ten trials. We also computed the accuracy and theory distance measures of FORTE's revised theories. The results are shown in Table 3. Overall, it appears that AUDREY II is slightly more accurate than FORTE, and AUDREY II's revised theories are closer to both the original theory and the known correct theory. However, these results must be considered tentative since they involve comparison with only a single run of FORTE. We believe the difference in distance between the two systems is accounted for by the fact that FORTE includes more complex operators that can produce more drastic changes than AUDREY II, and that AUDREY II contains an abduction mechanism that pinpoints where to generalize a theory.

Table 3: AUDREY II and FORTE on four KRK theories.

Accuracy		*Dist. to Original*		*Dist. to Correct*	
A-II	Forte	A-II	Forte	A-II	Forte
97.5	94.9	3.0	5	3.2	7
97.0	98.6	2.6	4	6.3	8
99.9	99.9	0.0	0	1.0	1
98.0	98.0	1.4	1	5.4	5

The difference between AUDREY II and FORTE is greatest for theory 1 since AUDREY II performs better on all three measures. The initial theory has three errors: two missing literals (in different clauses) and an extra clause. FORTE's revised theory deletes one of the clauses that was missing a literal, learns a semantically equivalent clause for the other missing literal, and specializes the extra clause into two clauses. AUDREY II's best revision of ten trials was semantically correct (100% accuracy) with distance three from the original and four from the correct theory. AUDREY II's worst revision was 92% accurate and only fixed one of the three errors.

Theory 2 had four errors: an incorrect clause, a missing clause, a clause missing two literals, and a clause with an incorrect literal. FORTE only fixed the missing clause by approximating it with three new clauses (this is why the distance measures are high). In every case, AUDREY II correctly found the missing clause and in four of the ten trials it learned an extra clause to repair some of the other errors. FORTE's revised theory outperformed AUDREY II's on accuracy but at the expense of creating a theory that was further from the original and correct theories than AUDREY II did on average.

Theory 3 had a single missing literal that had little effect on accuracy (the initial theory is 99.9% accurate). Neither FORTE nor AUDREY II repaired the theory so their performances on each dimension was the same.

Theory 4 had four errors: two missing literals, an extra clause that didn't cover any negatives, and an extra literal. FORTE only revised one of the clauses with a missing literal by deleting the clause. On this problem, FORTE and AUDREY II performed quite closely on all three measures of accuracy and distance.

6.3 Discussion

Since AUDREY II was designed to make small changes to theories, it performs significantly better than FOCL at theory revision. However, given the results of the previous section, it is clear that there is room for improvement of AUDREY II (and FORTE). The distance metric proposed here has shown us that although these systems make accurate revisions to theories, neither system al-

[4] There are actually five corrupt theories but the revised theory for the fourth case is incorrectly reported as the result for the fifth.

ways finds the syntactically correct concept definition when it is known.

One way to address this problem, used in KR-FOCL, is to allow the user to decide among changes that have the same accuracy. A user who knows that the part-time status of a student is a function of the number of credits the student is enrolled in, would not make the incorrect revision to the student loan problem reported in Section 6.1. A better approach might be to provide this knowledge to a theory revision system in the form of rule models [Davis, 1978] or higher-order relationships between predicates such as those expressible in CYC [Lenat and Guha, 1990].

7 Conclusion

Theory revision systems have the goal of improving the accuracy of the initial theory as well as some constraints on the type and degree of revisions that are desirable. We argue that revised theories should be as close to the original as possible. Unfortunately no theory revision systems to date have been evaluated in terms of the degree of revision made to the initial theory.

In this paper we proposed a measure for the distance between two theories. This measure corresponds to the number of edit operations at the literal level required to transform one theory into another. This measure can be used in conjunction with accuracy to give a better overall evaluation of the quality of revised theories when comparing one theory revision system to another. The distance metric can also be a useful tool when the correct theory is known. Comparisons of the revised and correct theories give a measure of a system's ability to learn theories in a form experts would write.

As further work is done in the area of theory revision, new techniques for evaluating and comparing systems are needed. We believe our measure of theory distance is a useful and important one and hope other researchers will adopt its use.

References

[Bergadano and Giordana, 1988] F. Bergadano and A. Giordana. A knowledge intensive approach to concept induction. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 305–317, Ann Arbor, Michigan, 1988. Morgan Kaufmann.

[Cohen, 1992a] W. Cohen. Abductive explanation-based learning: A solution to the multiple inconsistent explanation problem. *Machine Learning*, 8:167–219, 1992.

[Cohen, 1992b] W. Cohen. Compiling prior knowledge into an explicit bias. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 100–110, Aberdeen, Scotland, 1992. Morgan Kaufmann.

[Davis, 1978] R. T. Davis. Model-directed learning of production rules. In D. Waterman and F. Hayes-Roth, editors, *Pattern-directed inference systems*. Academic Press, 1978.

[Flann and Dietterich, 1989] N. Flann and T. Dietterich. A study of explanation-based methods for inductive learning. *Machine Learning*, 4:187–226, 1989.

[Hirsh, 1989] H. Hirsh. Combining empirical and analytical learning with version spaces. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 29–33, Ithaca, NY, 1989. Morgan Kaufmann.

[Kuhn, 1962] T. Kuhn. *The structure of scientific revolutions*. University of Chicago Press, 1962.

[Lawler, 1976] E. L. Lawler. *Combinatorial optimization: Networks and matroids*. Holt, Rinehart and Winston, 1976.

[Lenat and Guha, 1990] D. Lenat and R. V. Guha. *Building large knowledge-based systems: Representation and inference in the CYC project*. Addison-Wesley, 1990.

[Mooney and Richards, 1992] R. Mooney and B. Richards. Automated debugging of logic programs via theory revision. In *Proceedings of the International Workshop on Inductive Logic Programming*, Tokyo, Japan, 1992.

[Muggleton and Buntine, 1988] S. Muggleton and W. Buntine. Machine invention of first-order predicates by inverting resolution. In *Proceedings of the Fifth International Workshop on Machine Learning*, pages 339–352, Ann Arbor, MI, 1988. Morgan Kaufmann.

[Muggleton and Feng, 1990] S. Muggleton and C. Feng. Efficient induction of logic programs. In *Proceedings of the First Conference on Algorithmic Learning Theory*, Tokyo, Japan, 1990. Ohmsha.

[Norvig, 1992] P. Norvig. *Paradigms of artificial intelligence programming: Case studies in Common Lisp*. Morgan Kaufmann, 1992.

[Ourston and Mooney, 1990] D. Ourston and R. Mooney. Changing the rules: A comprehensive approach to theory refinement. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 815–820, Boston, MA, 1990. Morgan Kaufmann.

[Pazzani and Brunk, 1991] M. Pazzani and C. Brunk. Detecting and correcting errors in rule-based expert systems: An integration of empirical and explanation-based learning. *Knowledge Acquisition*, 3:157–173, 1991.

[Pazzani and Kibler, 1992] M. Pazzani and D. Kibler. The utility of knowledge in inductive learning. *Machine Learning*, 9:57–94, 1992.

[Quinlan, 1990] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[Richards and Mooney, 1991] B. Richards and R. Mooney. First-order theory revision. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 447–451, Evanston, IL, 1991. Morgan Kaufmann.

[Richards, 1992] B. Richards. *An operator-based approach to first-order theory revision*. PhD thesis, University of Texas, Austin, TX, 1992.

[Shasha and Zhang, 1990] D. Shasha and K. Zhang. Fast algorithms for the unit cost editing distance between trees. *Journal of Algorithms*, 11:581–621, 1990.

[Swartout, 1981] W. Swartout. Explaining and justifying in expert consulting programs. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 203–208, Vancouver, B.C., 1981. Morgan Kaufmann.

[Tai, 1979] K. C. Tai. The tree-to-tree correction problem. *J. Assoc. Comput. Mach.*, 26:422–433, 1979.

[Towell, 1991] G. Towell. *Symbolic knowledge and neural networks: Insertion, refinement and extraction*. PhD thesis, University of Wisconsin, Madison, WI, 1991.

[Wogulis, 1991] J. Wogulis. Revising relational domain theories. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 462–466, Evanston, IL, 1991. Morgan Kaufmann.