A Linguistically Based Semantic Bias for Theory Revision

Clifford Brunk and Michael Pazzani
Department of Information and Computer Science
University of California
Irvine, CA 92717
brunk@ics.uci.edu
pazzani@ics.uci.edu

Abstract

We present a new approach to theory revision that uses a linguistically based semantics to help detect and correct errors in classification rules. The idea is that preferring linguistically cohesive revisions will enhance the comprehensibility and ultimately the accuracy of rules. We explain how to associate terms in the rules with elements in a lexical class hierarchy and use distance within the hierarchy to estimate linguistic cohesiveness. We evaluate the utility of this approach empirically using two relational domains.

1 INTRODUCTION

Theory revision is the task of making a theory, which may contain errors, consistent with a set of examples. It is one of the tasks performed by a knowledge engineer during the creation of a rule-based expert system. Within the machine learning community there has been a focus on developing approaches to automate the task of using a set of examples to identify and repair errors in classification rules (e.g., SEEK2 (Ginsberg, Weiss & Politakis 1985), RTLS (Ginsberg 1990), EITHER (Ourston & Mooney 1990), DUCTOR (Cain 1991), AUDREY (Wogulis 1991), FORTE (Richards & Mooney, 1991), R_X (Tangkitvanich & Shimura 1992), and PTR+ (Koppel, Feldman & Segre 1994)).

Traditional theory revision approaches, including all those listed above, operate within a framework in which the semantics associated with a term in a rule is derived from the term's logical definition and the set of examples that definition satisfies. Invariably, these theory revision approaches use an example-based metric (e.g. accuracy, compression, or information gain) to guide the repair process. A limitation of relying solely on example-based metrics is that they often lead to repairs which, though accurate on the training set, combine terms that are not meaningful when used together. For instance in the student loan domain (Pazzani & Brunk 1991) it is not uncommon to see a revision system produce a rule that concludes a person is "enrolled-in-more-than-n-units" if that person is "unemployed" or "disabled". In spite of the fact that this rule is accurate on the training data, no competent English speaking knowledge engineer would create this rule. We believe the reason for this is that people operate in a richer semantic framework than current theory revision systems. Most English speaking people know that there is no connection between being "unemployed" or "disabled" and being "enrolled-in-more-than-n-units". In this paper, we present an extension to the semantic framework used by current theory revision systems that we believe will lead to more meaningful and ultimately more accurate revisions.

We propose augmenting the traditional framework to include a linguistically based semantics that associates the terms in the theory with elements in a lexical class hierarchy. Ideally, this hierarchy would encode all of the background knowledge of a knowledge engineer (c.f. Lenat & Guha 1990). Because a comprehensive encoding of human knowledge is not currently available, we have focused on using WORDNET (Miller 1990; Beckwith, et al. 1991), a lexical database containing the relationship between approximately 30,000 commonly occurring English words, as the foundation for our semantics. Since WORDNET already contains entries for terms that occur in commonly used problems (Student loans – Pazzani & Brunk 1991 and Moral Reasoning – Shultz 1990) we have been able to focus our efforts on the problem of how to use this information.

In the remainder of this paper, we present CLARUS, Concept Learning And Repair Using Semantics, an approach to revising first-order theories that utilizes linguistic information to guide the repair process. We will demonstrate that using a linguistically based semantic bias results in syntactically more desirable theories that tend to be more accurate than those produced without this bias.

2 BACKGROUND

CLARUS is an extension of the relational theory revision system A3. CLARUS uses WORDNET to prefer repairs based on the linguistic value of the resulting rule. In this section, we provide a brief review of A3 and WORDNET.

2.1 A3

A3 (Wogulis 1994) is a relational theory revision system that uses a set of examples to guide the repair process. In A3, the revision task is an iterative process of determining what to repair and how to repair it. A3 uses a single fault assumption mechanism to identify which relation within a theory to repair. An assumption is a goal-outcome pair. When an attempt is made to prove a goal that matches the goal portion of an assumption the goal succeeds or fails depending on the outcome portion of that assumption. For each misclassified example A3 records the set of single assumptions that allow the example to be correctly classified. After recording the assumptions for all misclassified examples, A3 selects the assumption which corrects the most errors. In the case of ties, the assumption used deepest in the call graph is selected.

The selection of an assumption targets the relation used in the assumption for repair. The type of repair required is determined by the outcome portion of the assumption and the context in which the assumption was used. In an unnegated context an assumption that forces a goal to fail indicates specialization is needed, while an assumption that forces a literal to succeed indicates generalization is required. In negated contexts the type of repair, specialization or generalization, is reversed.

Once the relation and the type of repair have been determined, A3 generates repair candidates by applying a set of operators to both the definition of the relation and to each clause containing a literal that uses the relation. The repair candidate yielding the largest increase in accuracy is retained. In the event that there is a tie between repair candidates, the repair that results in the theory with the smallest edit-distance to the initial theory is selected. The edit-distance between two theories is the number of single literal deletions, insertions, or replacements needed to transform one theory into the other. (Wogulis & Pazzani 1993)

2.2 WORDNET

WORDNET (Miller 1990; Beckwith et al. 1991) is a large lexical database, consisting of a set of data files and index files. The data files contain English words grouped into lexical classes called synonym sets or synsets for short. In addition to a collection of synonymous words each synset contains a set of pointers that describe the relation of that synset to other synsets. The semantic pointers found within a synset depend on what part of speech the synset represents. Figure 1, is a list of the WORDNET pointer types. The index files contain links between words and the synsets in which they occur. Within CLARUS, the index files are used to create a lexical tag for each relation in the theory, while the data files are used to determine the distance between lexical tags.

3 CLARUS

In this section we describe CLARUS. We show that the traditional semantic framework used for theory revision can be extended to include a linguistic bias. We begin by showing how to create lexical tags that link relations in the theory to synsets within the WORDNET lexical class hierarchy. Next, we explain how to measure distances between these tags. Finally, we describe how the lexical tags are used to estimate the linguistic value of a rule.

Used in CLARUS	
Hypernym	pointers to more general synsets.
Antonym	pointers to synsets that are in some sense opposite to the current synset.
Member-Holonym	pointers to synsets that represent a group of which the current synset is a member.
Substance-Holonym	pointers to synsets that represent the substance of which the current synset is composed
Part-Holonym	pointers to synsets that represent the whole of which the current synset is a part.
Similar-to/Also-see	pointers to synsets that are similar to the current synset.
Pertainnym	pointers to synsets representing the noun to which the current adjectival synset pertains.
Derived-from	pointers to the synsets representing the adjective from which the current synset is derived.
Not used in CLARUS	
Hyponym	pointers to more specific synsets.
Member-Meronym	inverse of *Member-Holonym.*
Substance-Meronym	inverse of *Substance-Holonym*
Part-Meronym	inverse of *Part-Holonym.*
Attribute	
Entailment	
Cause	

Figure 1: WORDNET pointer types

```
(((V 0734246) or (V 0723002) or (R 0034640) or … or (A 0062687)) and        ;; longest
 ((N 6347345) or (N 0421139)) and                                           ;; absence
 () and                                                                     ;; from
 ((V 0945155) or (N 6649549) or (N 4005100) or … or (N 2364547)))           ;; school
```

Figure 2: The lexical tag for the relation `longest-absence-from-school`. Each line is a disjunction of pointers to the synsets representing a word in the relation name. Items such as `(V 0734246)` are pointers to synsets. Those starting with `V` refer to verbs, `A` adjectives, `N` nouns, and `R` adverbs.

3.1 GENERATING LEXICAL TAGS

A lexical tag provides a connection between the logical definition of a relation and an alternate view of the relation based on lexical classes. The lexical tag for a relation is generated by finding the words contained in the relation's name and matching them against the entries in the WORDNET index files. The relation name is processed from left to right. The longest underscore or hyphen delimited prefix of the relation name that matches an entry in a WORDNET index file is removed and the associated disjunction of synset pointers from the index file is conjoined to the relation's lexical tag. Processing continues until the entire relation name has been converted. Figure 2 is an example of the lexical tag created for the relation `longest-absence-from-school`.

The lexical tag itself is a conjunction of disjunctions of synset pointers. The conjunction represents the meaning of the entire relation name. Each disjunction represent the meaning of one word in the relation name. A disjunction is used because there can be many possible meanings for each word. Each disjunction contains all the synsets associated with the word it represents.

It would be possible for the user to manually create the lexical tag for a relation. By choosing the synset that is closest to the intended meaning of each word in the relation name, the user could specify a rather precise meaning for each relation. Although manual lexical tag generation is possible, in this paper we focus on the automatic lexical tag generation approach outlined above and present results obtained using that technique.

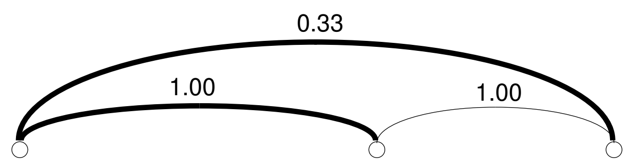
3.2 LINGUISTIC HETEROGENEITY

CLARUS uses the lexical tag associated with each relation to estimate the linguistic value of the repair candidates. We have developed a measure we call *linguistic heterogeneity* that is an estimate of the linguistic value of a rule. In CLARUS, repairs that produce rules with lower linguistic heterogeneity are preferred to those that produce rules with higher linguistic heterogeneity. The linguistic heterogeneity of a rule is not directly related to the set of examples the rule covers and therefore provides an independent evaluation criterion.

In CLARUS, a rule is a set of PROLOG clauses with the same head. Each clause is composed of a head and a body. The head is a literal that can be deduced to be true if every literal in the body is true. Consider the clause in Figure 3, it states that if a person (e.g., `sue`) is enlisted in the navy (`enlist(sue navy)`), and the navy is a branch of the armed-forces (`armed-forces(navy)`), then she is eligible for a military-deferment (`military-deferment(sue)`).

The linguistic heterogeneity of a clause can be expressed in terms of a minimum weight spanning tree. Each literal in the clause can be thought of as a vertex. There is an edge between each pair of literals. The weight associated with an edge is the lexical distance between the literals it connects. The linguistic heterogeneity of a clause is the weight of the minimum weight spanning tree connecting each literal in the clause. In Figure 3, the edges forming the spanning tree appear in bold. The spanning tree approach has the following desired properties:

1. Clauses with more closely related literals tend to be less linguistically heterogeneous than clauses of equal length with more distantly related literals.
2. Adding a literal to a clause that bridges the gap between other literals can reduce the linguistic heterogeneity of the clause.
3. Shorter clauses tend to be less linguistically heterogeneous than longer clauses.




```
military_deferment(?P) :- enlist(?P ?O), armed_forces(?O).
```

Figure 3: Spanning tree for a clause with a linguistic heterogeneity of 1.33.

	Distance Between	Example	Value
LD_r	relation names	(military-deferment , peace-corps)	0.83
D_{lt}	lexical tags	((((A 1280639) or ... or (N 3975222)) and ((N 0361434))) , (((N 3974151))))	0.83
D_d	disjunctions of synsets	(((A 1280639) or ... or (N 3975222)) , ((N 3974151)))	0.75
D_s	synsets (*normalized*)	((N 3975222) , (N 3974151))	0.75
D	synsets	((N 3975222) , (N 3974151))	3

Figure 4: Distance Functions and Examples

We define the linguistic heterogeneity of a rule to be the sum of the linguistic heterogeneity of its constituent clauses. Therefore, the linguistic heterogeneity of a rule increases as the number of clauses needed to define that rule increases. The linguistic heterogeneity of a clause is the sum of the edge weights of the minimum weight spanning tree connecting each literal in the clause where the weight is the lexical distance between literals.

3.3 LEXICAL DISTANCE

The lexical distance between two literals is the lexical distance between the relation names of each literal. The lexical distance between two relation names,

$$LD_r(r_1,r_2) = D_{lt}\big(lexical\_tag(r_1), lexical\_tag(r_2)\big),$$

is the distance between the lexical tag associated with each relation name. Recall that a lexical tag is a conjunction of disjunctions of synset pointers. The distance between two lexical tags, lt_1 and lt_2 , is calculated as follows. For each disjunction, d_i, in lt_1 find the disjunction, d_j, in lt_2 which is closest to d_i. Then, for each disjunction, d_j, in lt_2 find the disjunction, d_i, in lt_1 which is closest to d_j. We define the average of these distances to be,

$$D_{lt}(lt_1,lt_2) = \frac{\sum_{d_i \in st_1} \underset{d_j \in st_2}{Min}\big(D_d(d_i,d_j)\big) + \sum_{d_j \in st_2} \underset{d_i \in st_1}{Min}\big(D_d(d_j,d_i)\big)}{|lt_1| + |lt_2|}.$$

$D_{lt}(lt_1,lt_2)$ ranges from 0 to 1 inclusive, 0 indicates the lexical tags are very similar while 1 indicates they are very dissimilar. We define the distance between two disjunctions of synsets to be the minimum of all pairwise distances between the synsets,

$$D_d(d_1,d_2) = \underset{\substack{s_i \in d_1 \\ s_j \in d_2}}{Min}\big(D_s(s_i,s_j)\big).$$

$D_d(d_1,d_2)$ ranges from 0 to 1 inclusive, 0 indicates that d_1 and d_2 have at least one synset in common while 1 indicates that no synset in d_1 is related to any synset in d_2 Resnik (1993) argues that using the minimum gives an overly optimistic estimate of lexical closeness; instead he proposes using the average of the pairwise distances. We have yet to encounter a problem using the overly optimistic minimum estimate.

All of the lexical distance estimates are based on the distance between WORDNET synsets. We define the distance between two synsets, $D(s_1,s_2)$, to be the minimum number of pointers that have to be traversed in order to reach the nearest synset linked to both s_1 and s_2. We always count traversing pointers of the following types: *Hypernym*, *Holonym*, *Pertainnym*, and *Derived-from*. In addition, we count traversing *Antonym* and *Similar-to/Also-see* as the initial step from s_1 and s_2. When two synsets are not linked to a common synset, we define the distance between them to be infinite. Because $D(s_1,s_2)$ ranges from 0 to infinity, we simplify calculations by using the normalized distance between synsets,

$$D_s(s_1,s_2) = 1 - \frac{1}{1 + D(s_1,s_2)}.$$

Figure 4 is a summary of the notation we have used to describe lexical distances, plus an example of each. The values displayed are the values calculated automatically from WORDNET.

Measuring the distance between lexical tags in this way matches our intuition. As can be seen in Figure 5, the distance between synonymous concepts like `military` and `armed-forces` is minimal (0.00), the distance between closely related concepts like `army` and `military` or `nickel` and `dime` is small (0.50), the distance between less related concepts like `military` and `peace-corps` is larger (0.75), and the distance between unrelated concepts like `nickel` and `military` is maximal (1.00).

	armed-forces	**army**	**dime**	**military**	**nickel**	**peace-corps**
armed-forces	0.00					
army	0.50	0.00				
dime	1.00	1.00	0.00			
military	0.00	0.50	1.00	0.00		
nickel	1.00	1.00	0.50	1.00	0.00	
peace-corps	0.75	0.80	1.00	0.75	1.00	0.00

Figure 5: WORDNET based estimates of lexical distance between relations.

3.3 A LINGUISTIC BIAS

In CLARUS, linguistic heterogeneity is used as a bias when determining which relation to repair and what repair to apply to that relation. Unlike A3, which breaks ties between competing repairs of equal accuracy using the edit-distance to the initial theory, CLARUS breaks ties by preferring the repair that leads to the theory with the lowest linguistic heterogeneity. This is computed by calculating the change in linguistic heterogeneity produced by any change to the theory.

CLARUS differs from A3 in one other way. A3 uses its assumption mechanism to find the single relation that could most increase accuracy. Once this relation has been selected, the repair operators are applied and the best repair is selected, even if that repair is not as good as the assumption mechanism's estimate. This approach is not well suited to performing linguistic comparisons, because it doesn't compare competing repairs to different relations. Instead, it focuses on evaluating competing repairs to the same relation. On the other hand, CLARUS uses a more extensive search strategy, like that of FORTE (Richards, 1991). CLARUS uses A3's assumption mechanism to order the relations according to an upper bound on the potential increase in accuracy a modification to that relation could have. The A3 repair operators are applied to each relation until a repair is found that is more accurate than the estimated potential of any change to any subsequent relation. This search strategy is more computationally expensive than that of A3. But, it allows CLARUS to frequently compare competing repairs to different relations allowing the linguistic bias to have more of a role in determining the quality of repairs.

4 RESULTS

In the real world, a theory revision system would be used to assist in the creation and refinement of a theory that describes a set of examples. We would like to show that CLARUS makes repairs similar to those made by a human knowledge engineer when debugging a prototype knowledge-based system on a database of cases. But, it is impractical to evaluate theory revision techniques in this context, because it requires presenting a group of knowledge engineers with a variety of erroneous theories and cases. Instead, we adopt a methodology whereby errors are introduced into a correct theory and the revision system's ability to repair the mutated theory is evaluated. Ideally, a theory revision system would undo the errors introduced to the correct theory. We measure the edit-distance from the revised theory to the correct theory to quantify the degree to which this ideal has been achieved.

In this section, we report on a series of experiments in which we empirically evaluate the utility of the linguistic bias. Our hypothesis is that using this bias will tend to decrease the distance between the repaired theory and the correct theory, and may offer a small improvement in accuracy when compared to theory revision systems that don't use this bias.

4.1 STUDENT LOAN

We start by examining the differences in the repairs produced by A3 and CLARUS. Consider the student loan domain (Pazzani & Brunk 1991). The theory in this domain indicates when an individual is not required to make student loan payments.

no_payment_due(?P) :- continuously_enrolled(?P).

no_payment_due(?P) :- eligible_for_deferment(?P).

continuously_enrolled(?P) :- enrolled_in_more_than_n_units(?P 5), ~~never_left_school(?P).~~

never_left_school(?P) :- longest_absence_from_school(?P ?M), >(6 ?M).

eligible_for_deferment(?P) :- military_deferment(?P).
eligible_for_deferment(?P) :- peace_corps_deferment(?P).
eligible_for_deferment(?P) :- financial_deferment(?P).
eligible_for_deferment(?P) :- student_deferment(?P).
eligible_for_deferment(?P) :- disability_deferment(?P).

military_deferment(?P) :- enlist(?P ?O), armed_forces(?O).

peace_corps_deferment(?P) :- enlist(?P ?O), peace_corps(?O).

~~financial_deferment(?P) :- unemployed(?P).~~
financial_deferment(?P) :- enrolled(?P UCI ?U).
financial_deferment(?P) :- filed_for_bankruptcy(?P).

student_deferment(?P) :- enrolled_in_more_than_n_units(?P 11).

disability_deferment(?P) :- **not(male(?P)),** disabled(?P).

enrolled_in_more_than_n_units(?P ?N) :- enrolled(?P ?S ?U), school(?S), >(?U ?N).

Figure 7: Corrupted student loan domain theory (Pazzani & Brunk, 1991). Strike-through indicates clause or literal deletion and bold indicates clause or literal addition.

```
no_payment_due(?P) :- continuously_enrolled(?P), never_left_school(?P).
financial_deferment(?P) :- enrolled(?P UCI ?U), disabled(?P).
enrolled_in_more_than_n_units(?0 ?1) :- unemployed(?0).
enrolled_in_more_than_n_units(?0 ?1) :- disabled(?0).
```

Figure 8: A3 Repairs.

```
no_payment_due(?P) :- continuously_enrolled(?P), never_left_school(?P).
disability_deferment(?P) :- ~~not(male(?P))~~, disabled(?P).
financial_deferment(?P) :- ~~enrolled(?P UCI ?U),~~ unemployed(?P).
```

Figure 9: CLARUS Repairs.

Figure 7 contains a correct theory that has been intentionally corrupted. Figure 8 contains the repairs that A3 makes to this theory when given 25 examples of people who are required to make loan payments and 25 examples of people who are not. Although the A3 repairs result in a theory that correctly classifies all examples, not one of the errors introduced to the correct theory has been undone. Instead additional modifications have been introduced. These modifications combine linguistically unrelated relations. For instance, A3 creates a rule that concludes a person is `enrolled-in-more-than-n-units` if that person is `unemployed` or `disabled`. A3 also creates a rule that concludes a person is eligible for a `financial-deferment` if that person is `enrolled` at `UCI` and `disabled`. It is unlikely that a knowledge engineer would combine these relations to form rules. Nonetheless, these are precisely the kinds of repairs made by A3, FORTE and most other theory approaches.

CLARUS operates in a richer semantic framework than other approaches. It uses linguistic clues provided by the relation names in addition to the training examples to guide the revision process. Figure 9 contains the repairs that CLARUS makes to the theory in Figure 7. CLARUS fixes 3 of the 4 mutations introduced to the correct theory. In addition the repairs made by CLARUS are similar to the kind of repairs that a knowledge engineer might make. For instance, CLARUS creates a rule concluding that a person is eligible for a `financial-deferment` if that person is `unemployed`, and CLARUS removes the condition excluding a `male` from being eligible for a `disability-deferment`. We feel that these repairs are more desirable than those produced by A3, because they result in a theory which is syntactically very similar to the correct theory and because they result in clauses that are linguistically more meaningful than those produced by A3.

A detailed comparison of the repairs made by A3 and CLARUS to a single theory provides some insight into the approaches, but it does not show that the extended semantic framework helps in a wide variety of contexts. To substantiate our hypothesis, we perform an experiment using four related algorithms: A3, Semantic-A3 (A3 using linguistic heterogeneity to break ties in accuracy), Non-Semantic-CLARUS (a version of CLARUS that breaks ties in accuracy randomly) and CLARUS. Non-Semantic-CLARUS uses the extended search for repairs like FORTE, but doesn't use linguistic heterogeneity to break ties.

We use 1000 randomly generated examples of the student loan relation partitioned into disjoint training and testing tests. The training sets range from 10 to 100 examples, while the testing set is fixed at 500 examples. Mutations are introduced into the correct theory using four operators: add-literal, add-clause, delete-literal and delete-clause. Literals are created by selecting a random relation from the theory and creating a list of randomly selected variables that are consistent with the type constraints of the relation. During each trial a number of random mutations are introduced into the correct theory, this mutated theory is used to generate a revision curve for each algorithm. The revision curve is generating my having each algorithm attempt to repair the theory, using 10, 25, 50, 75 and 100 examples. Each algorithm is presented with the same initial theory and the same training examples. The repaired theories are all evaluated on the same 500 test examples.

Figure 10 and 11 show the results from this experiment averaged over 100 trials. In Figure 10, we look at the accuracy of each algorithm. For the most part, the linguistic bias makes little difference in accuracy, although extra search does tend to produce more accurate rules. In Figure 11, we look at the edit-distance between the revised-theory and the correct-theory. There is a significant difference between Semantic-A3 and A3 at least at the .05 level when there are more than 10 training examples as determined by a paired two-tailed t-test. Similarly, the CLARUS and A3 edit-distance curves are also significantly different for every training set size above 10. In summary, we see that using the WORDNET based linguistic bias does significantly decrease the edit-distance between the correct and revised theories. Although, it does not significantly effect accuracy, it appears to offer a slight advantage to CLARUS.

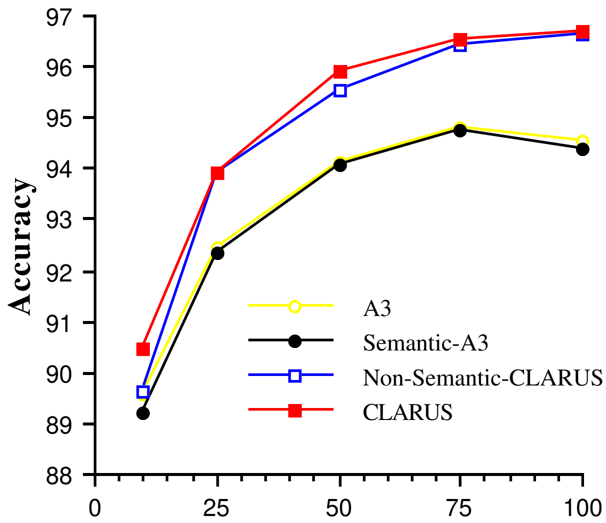


Figure 10: Accuracy as a function of the number of training examples.

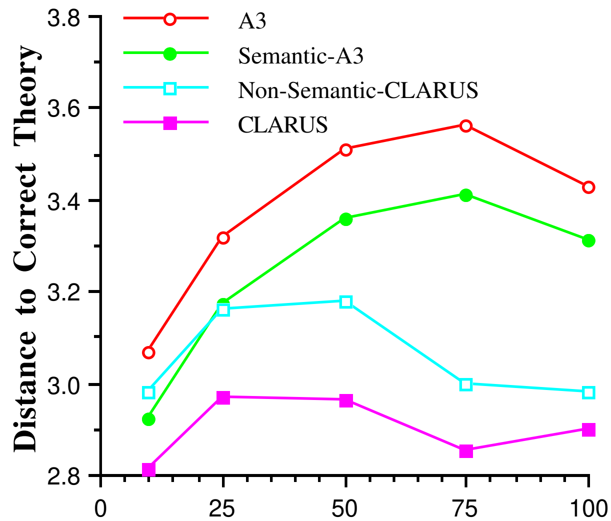


Figure 11: Edit-distance as a function of the number of training examples.

In a second experiment on the student loan we demonstrate that CLARUS, a theory revision approach, is competitive with purely inductive systems like FOIL (Quinlan 1990) and theory-guided inductive systems like FOCL (Pazzani, Brunk & Silverstein 1991; Pazzani & Kibler 1992). In this experiment we compare CLARUS, FOCL and EMPIRICAL. EMPIRICAL is FOCL's inductive component, a reimplementation of FOIL that handles relations defined as prolog rules as well as relations defined extensionally.

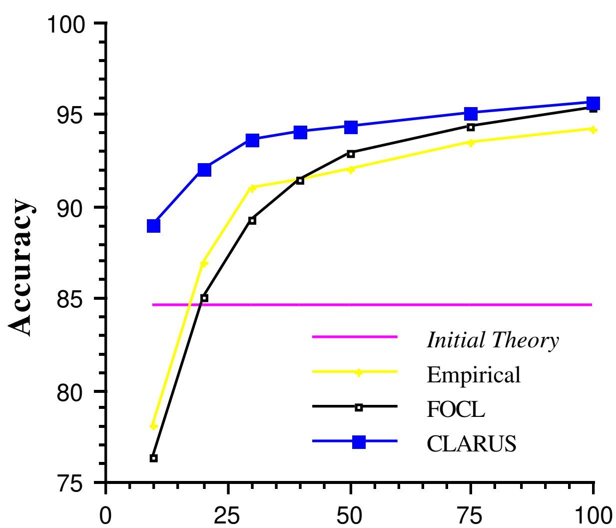


Figure 12: Accuracy as a function of the number of training examples.

Following the same methodology used earlier we run paired tests. Each algorithm is given the same training data, and CLARUS and FOCL, are given the same initial theory. The results in this experiment, are graphed in Figure 12. CLARUS is significantly more accurate than the other algorithms when there are between 10 and 50 training examples, and never less accurate then the others with greater numbers of training examples.

4.2 MORAL REASONER

The Moral Reasoner (Shultz 1990; Darley & Shultz 1990) is a rule-based model that qualitatively simulates how a person reasons about harm-doing. We used the theory as presented in Wogulis (1993) with two modifications[1] which facilitate the lexical tagging process. We use the same methodology used in the first experiment on the student loan domain. We compare A3, Semantic-A3, Non-Semantic-CLARUS and CLARUS in paired tests using 208 examples, 102 positive and 108 negative, partitioned into disjoint training and testing tests. The training sets range from 10 to 100 examples, while the test set is fixed at 100 randomly selected examples.

Figure 13 and 14 are the results from this experiment averaged over 50 trials. In Figure 13, we look at the accuracy of each algorithm. In Figure 14, we look at the edit-distance between the revised-theory and the correct-theory.

---

[1] We corrected the spelling of the relation name `reponsible` so that it now appears as `responsible`, and we changed the relation name `weak_intend1` to `weak_intend_1`. In the future, we will extend the relation name parser in CLARUS to allow numbers as delimiters between words.

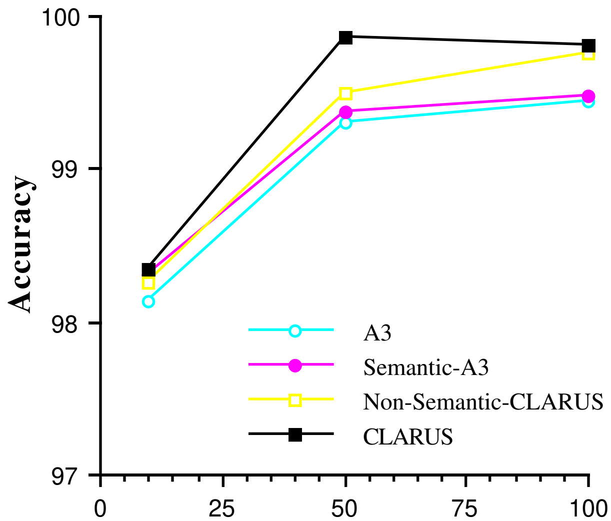


Figure 13: Accuracy as a function of the number of training examples.

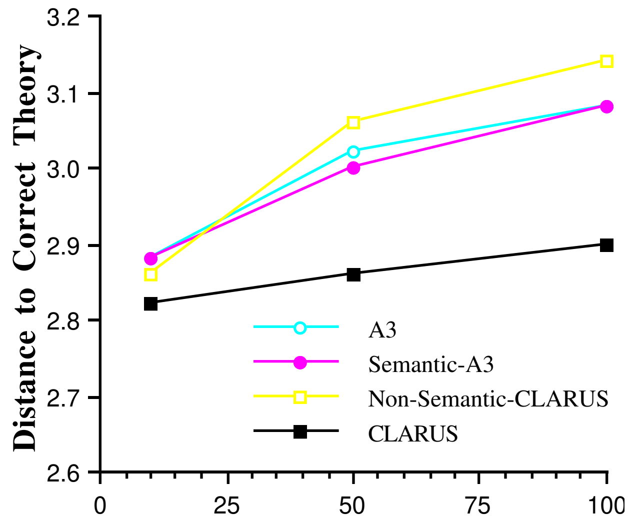


Figure 14: Edit-distance as a function of the number of training examples.

As in the student loan domain the trend is for the linguistic bias to decrease the edit-distance between the revised and correct theory. This is clearly seen by the significant difference between Non-Semantic-CLARUS and CLARUS. However, when comparing A3 and Semantic-A3 the difference is not significant. This tends to support our belief that the extended search performed by CLARUS enhances the effect of the linguistic bias. There is not a statistically significant difference between the accuracy of any of the algorithms, although as in the student loan domain, the linguistic bias appears to offer CLARUS a slight advantage in accuracy for CLARUS.

5 Discussion

The empirical evidence provided by the student loan and moral reasoner domains offer compelling support for our hypothesis that using a linguistically based semantic bias decreases the edit-distance between the revised and correct theory. This provides evidence that the extended semantic framework helps achieve our primary objective of biasing a theory revision system to produce repairs that a human knowledge engineer would consider plausible. There is also weaker evidence that the linguistic bias can help increase accuracy slightly. We have begun to investigate the affect of the linguistic bias on the accuracy of intermediate concepts.

CLARUS is a promising approach, but it has limitations. The theory to be repaired must contain terms with descriptive names. Without these names, the linguistic-bias offers no assistance in the revision process. For example, WORDNET does not contain terms that are useful in the promoter theory (Towell 1991) or the King-Rook-King chess theory (Muggleton, Bain, Hayes-Michie & Michie 1989). In both the student loan and moral reasoner domain, the semantics in WORDNET are complete enough to provide a useful bias.

We have presented CLARUS using WORDNET as the basis for the linguistic semantics. As other sources of lexical and semantic knowledge become available, for instance CYC (Lenat & Guha 1990), the CLARUS approach will become even more feasible. We chose WORDNET because it was easy to acquire, via anonymous FTP, and simple to interface with. However, it is not clear that the data contained in WORDNET is ideally suited to the theory revision task. It does not contain connections between some concepts that we would have thought natural. For example, in WORDNET there is no connection between the synsets for "disabled" and the synsets for "disability". A solution to this problem is to manually generate the lexical tags associated with the terms in the theory. This would allow the knowledge engineer to create a precise linguistic description for each term, rather than attempting to generate one automatically. We have preliminary results that indicate, not surprisingly, that manually created lexical tags result in better repairs than automatically generated lexical tags. We have also started exploring an alternative to WORDNET in which a user enters the semantic distance between relation names directly, rather than computing them as we have in this paper.

There are many approaches related to CLARUS. We have already described its relation to A3, yet we have not pointed out that the underlying learning mechanism in CLARUS is FOCL (Pazzani & Kibler 1992), which in turn is based on FOIL (Quinlan 1994). CLARUS also borrows ideas from FORTE (Richards 1992), specifically its search strategy, but CLARUS uses a simpler set of repair operators than FORTE. Additionally, CLARUS inherits from A3 a strategy that deals with negation in a more complete manner than FORTE. However, the novel contribution of the CLARUS approach is the realization that the framework in which theory revision is traditionally performed can be extended to include a linguistically based semantics. We believe that this extended semantic framework would also help other theory revision systems such as EITHER and FORTE to produce theories that are more similar to those produced by a human knowledge engineer.

6 CONCLUSION

Recently, one of the unique aspects of machine learning has been its focus on knowledge-intensive approaches. CLARUS continues this line of investigation. CLARUS makes use of background knowledge in the form of an initial theory and in the form a lexical database. We have shown in this paper that the lexical semantics can bias a theory revision system to make repairs that appear more meaningful than those of prior approaches.

Acknowledgments

We would like to thank Kamal Ali, Chris Merz, Piew Datta, and Lisa Ferrendelli for their comments and suggestions about this research. Thanks are also do J.R. Quinlan, James Wogulis, and George Miller for providing a foundation upon which to build.

References

Beckwith, R., Fellbaum, C., Gross, D., & Miller, G. (1991). WordNet: A lexical database organized on psycholinguistic principles. *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, ed. Zernik, U. (211–232). Lawrence Erlbaum.

Cain, T. (1991). The DUCTOR: A Theory Revision System for Propositional Domains. *Proceedings of the Eighth International Workshop on Machine Learning* (485–489). Evanston, IL: Morgan Kaufmann.

Ginsberg, A., Weiss, S., & Politakis, P. (1985) SEEK2: A Generalized Approach to Automatic Knowledge Base Refinement. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*. (367–374). Los Angeles, CA: Morgan Kaufmann.

Ginsberg, A. (1990) Theory Reduction, Theory Revision, and Retranslation. *Proceedings of the Eighth National Conference on Artificial Intelligence* (777–782).

Koppel, M., Feldman, R., & Segre, A., (1994). Bias-Driven Revision of Logical Domain Theories. *Journal of Artificial Intelligence Research*, 1, (1–50). AI Access Foundation and Morgan Kaufmann.

Lenat, D., & Guha, R. (1990). *Building Large Knowledge Base Systems.* Reading, MA: Addison Wesley.

Miller, G. (1991). WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4).

Muggleton, S., Bain, M., Hayes-Michie, J., & Michie, D. (1989). An experimental comparison of human and machine learning formalisms. *Proceedings of the Sixth International Workshop on Machine Learning* (pp. 113–118). Ithaca, NY: Morgan Kaufmann..

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

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

Pazzani, M., Brunk, C., & Silverstein, G. (1991). A knowledge-intensive approach to learning relational concepts. *Proceedings of the Eighth International Workshop on Machine Learning* (32–436). Evanston, IL: Morgan Kaufmann.

Pazzani, M., & Kibler, D. (1992). The utility of knowledge in inductive learning. *Machine Learning*, 9, (57–94).

Quinlan, J.R., (1990). Learning logical definitions from relations. *Machine Learning*, 5, (239–266).

Richards, B. (1992). An Operator-Based Approach to First-Order Theory Revision. Ph.D. Thesis. University of Texas, Austin.

Richards, B. & Mooney, R. (1991). First-Order Theory Revision. *Proceedings of the Eight International Workshop on Machine Learning* (447–451). Evanston, IL: Morgan Kaufmann.

Resnik, P. (1993). Selection and Information: A Class-Based Approach to Lexical Relationships. Ph.D. Thesis. University of Pennsylvania.

Shultz, T. (1990). A rule based model of judging harm-doing. *Proceedings of the Twelfth Annual Conference on the Cognitive Science Society* (229–236). Cambridge, MA: Lawrence Erlbaum.

Tangkitvanich, S., & Shimura, M. (1992) Refining a Relation Theory with Multiple Concepts and Subconcepts. *Proceedings of the Ninth International Workshop on Machine Learning,* (436–444). Aberdeen, Scotland: Morgan Kaufmann.

Towell, G. (1991). *Symbolic Knowledge and Neural Networks: Insertion, Refinement, and Extraction*. Ph.D. Dissertation, Computer Sciences Department, University of Wisconsin.

Wogulis, J. & Pazzani, M. (1993). A methodology for evaluating theory revision systems: Results with AUDREY II. *The International Joint Conference on Artificial Intelligence,* (1128–1134), Chambery, France.

Wogulis, J. (1994) An Approach to Repairing and Evaluating First-Order Theories Containing Multiple Concepts and Negation. Ph.D. Thesis. University of California, Irvine.