Explanation Based Learning as Constrained Search*

David Haines
Experimental Knowledge Systems Laboratory

COINS Technical Report 89-33

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

April 1989

Abstract

Explanation Based Learning requires a great deal of information in order to create proofs. This paper suggests that the use of proofs in EBL is primarily to constrain search. Therefore an alternate approach for EBL is to use non-provable explanations to constrain search and use performance testing to discriminate between the resulting generalizations. The approach is determined in the domain of English word pronunciation.

*This research is supported by the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research contract #N00014-87-K-0238, the Office of Naval Research, under University Research Initiative Grant Contract #N00014-86-K-0764, and NSF Presidential Young Investigators Award NSFIST-8351863.

1 Introduction

Explanation Based Generalization [Mitchell *et al.*, 1986] and Explanation Based Learning [Dejong and Mooney, 1986] (both referred to as EBL hereafter) have become popular techniques because they require few examples and allow the use of domain information. By a deep analysis of a single example EBL can make significant generalizations.

EBL, however, requires a great deal of domain knowledge. This paper argues that even in a domain where a complete domain theory may be impossible, EBL techniques can be helpful in creating good generalizations.

1.1 EBL

The classical method of EBL [Mitchell *et al.*, 1986] is to take take a single example of some concept and create a proof that the example is an instance of that concept. The logical form of this proof is then used to infer a re-expression of the concept in the representation language of the examples. EBL creates a new description of the concept in the efficient, operational, terms of the examples in the domain. This new representation is more general than the original example because aspects of the example that don't figure in the proof don't appear in the generalization. It is more efficient than the original concept because of its operational representation.

The dependence on proof makes it difficult to extend EBL to deal with imperfect domains. However this extension is both necessary and possible. Not only are theories hard to build [Rajamoney, 1988] but there are also domains where there appears to be no good complete domain theory. English word pronunciation, for example, is quite irregular. No one has been able to find a comprehensive theory to explain how it works (see section 2.1).

1.2 EBL as Constrained Search

EBL may not at first seem to be a search process; there is no search for generalizations as we have them in hand in the proof tree. All of the stages of the proof have been justified by the use of truth preserving inference. We may search among these valid generalizations for the most useful ones, but we don't need to search for correct ones.

However we can view this differently: EBL doesn't search for generalizations because it uses truth preserving inference to trivialize the generalization search space. In this view a proof is valuable because it constrains the concepts that will be considered. Generalizations outside of the proof aren't considered as they have some information that wasn't essential to the proof.

EBL involves search, but search has shifted to the proof process.

This view suggests two alternatives when an inadequate domain theory breaks the typical EBL scenario. On one hand we may wish to keep truth preserving inference and try to enhance the theory to allow for proofs. On the other hand we may wish to use a different inference method; one that is not truth preserving (and does not do proofs), but still uses explanations to constrain search. A new source of inferences is not a license to wild or random inference, just a loosening of the bounds that will now allow plausible or reasonable inference instead of just "true" inference.

This second approach has the advantage that it doesn't require complete knowledge to create the explanations. We also create the opportunity for using multiple explanations. Differing explanations will constrain search in different ways and to different amounts. This is both an advantage (we don't need the correct theory) and a disadvantage (we are likely to get somewhat poorer performance). When we give up truth preserving inference, we get the possibility of producing incorrect inferences. Therefore work taking this approach needs to propose some method to control this error.

This research is most similar in spirit to Dejong and Mooney [1986]. The approaches are similar in that they see explanations as methods to constrain and structure search. This current work differs in not requiring explanations to be causal and in suggesting the use of multiple explanations for a single example.

1.3 Other Relevant Research

Most of the current research efforts into the EBL domain theory problems are examples of the theory enhancement approach.

In ADEPT [Rajamoney, 1988] explanations that lead to poor performance are classified according to the way that they fail. To correct the failure of the domain theory extensions to the theory are created, and then experiments are proposed to discriminate between the different proposed extensions.

Clancey [1988] describes a method for finding gaps in a domain theory and then creating questions to be answered by a teacher or by experimentation. The answers to these questions provide an evaluation of the proposed changes.

In SIERRA [1987] VanLehn uses almost complete explanations to allow inferring that an extension to the theory is required. If an explanation is almost sufficient, and the information needed to make it sufficient can be identified then that information can be added to the domain theory.

All of this work relies on there being some good domain theory that is, if not correct, at least good enough to produce an almost correct explanation. It assumes that one can get an explanation that can be patched to become the correct explanation. Little work (besides this work on MOB) has dealt with the problem of working in domains where there is thought to be no complete domain theory.

2 The Mob System

MOB is a failure-driven supervised learning system whose task is the pronunciation of novel words. Each training instance is a correct word/pronunciation pair. When MOB is given a pair that contains a word that it fails to pronounce correctly MOB attempts to learn the correct pronunciation.

2.1 Domain

The domain of MOB is English word pronunciation. This is a particularly suitable area for exploring problems with insufficient domain theories because there is no known complete theory of how to map letters onto phonemes and generate the correct pronunciations [Stanfill and Waltz, 1986]. English pronunciation has clear regularities. Double consonants sound only once (APPLE). An E on the end of a word often indicates that the vowel earlier in the word is long, but the E is silent (HOME, TOME, MORE, HIKE, GRATE). However there are many exceptions to such regularities. SOME is not pronounced with a long O and a silent E. THOUGH and TOUGH share no phonemes even though the only difference is the added H in THOUGH. We don't have, and don't expect to have, a complete or consistent domain theory for word pronunciation.

2.2 Explanations and Inference in MOB

If there is no well defined domain theory for word pronunciation, what then can be good explanations for why a given sequence of letters should be pronounced in a particular way?

Rather than rely on finding the putative *right* explanation MOB creates a number of possible explanations; each a different way that the pronunciation of a word could be justified. There is no apriori way to discriminate between them and decide which is the best explanation; they are all candidates.

2.3 Explanations and Generalizations

The basic method used by MOB is to build an explanation tree (see figure 1). The tree gives a decomposition of a word/pronunciation pair into a number of mappings from subsequences of the letters to subsequences of phonemes. These mappings can be treated as rules for how to pronounce this sequence of letters. MOB is successful if it gathers a set of these rules that pronounces words accurately.

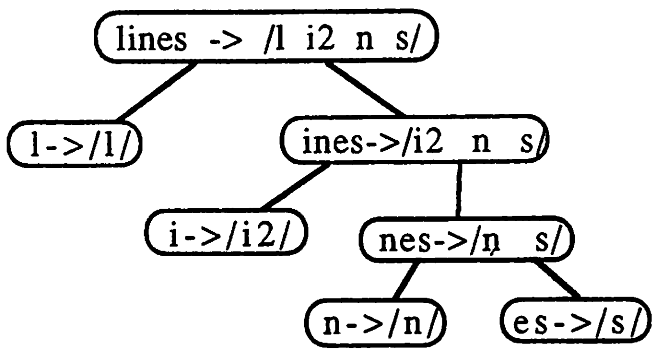


Figure 1: *An Explanation Tree*

Each explanation tree gives a number of sets of rules that can be used to correctly pronounce this training word. In figure 1 we can find, among others, the single rule {LINES→/l i2 n s/} or the set of rules {L→/l/, I→/i2/, N→/n/, ES→/s/}. Both are potential explanations, the first if the pronunciation is highly idiosyncratic, the second if it is highly regular.

We can see these rules as generalizations, rules to apply in the general situation of novel words; therefore a single explanation tree gives rise to a number of different possible generalizations. It may be that the most general rules, in the context of the prior rules learned, won't work; they may be too general or there may be no way to decide between two rules that map the same letters into different phonemes. This may require using more specific generalizations, ones closer to the root of the explanation tree. For example I→/i2/ may not be a good rule in general. It may be that a more useful mapping is the more specific INES→/i2 n s/.

This representation of explanations is instantly operational. It doesn't keep the explanation justifications explicitly, but only keeps around the operational version of each step.

This explanation representation also produces the possible generalizations as a side effect. Conceptually this generalization method is *context regression*. A very general mapping is refined by adding more context to it. The leaves of the explanation tree provide a set of very general mappings from letters to phonemes. Other, more specific, generalizations are formed by combining with the local context, in this case neighboring nodes. Since nodes in different explanation trees that are formed by different methods have different neighbors, MOB's technique of using multiple explanations results in a variety of generalizations.

2.4 Creating Explanations

Explanation trees are created by a grammar that specifies acceptable ways to map letters to phonemes. The parse tree of the grammar gives the structure of the explanation tree. The contents of each node are formed by the generalization method. Specifically in MOB the grammar is used to construct an explanation tree where each node represents a mapping from letters in the word to phonemes in the pronunciation (see figure 1). Each grammar is ambiguous and so will provide a number of different possible explanations. MOB also uses a number of these grammars at the same time, so that different types of explanations will be produced.

The grammars used in MOB are show in figures 2 and 3. They both contain nonsyntactic constraints on how the tree should be constructed. These constraints limit the number of explanations generated. Having these two degrees of freedom (multiple grammars and ambiguous grammars) means that MOB creates a number of explanations, none of which are known to be more or less plausible that another. All are used to propose generalizations. Testing is used to discriminate between the generalizations that the theory could not discriminate.

X→S
X→U
X→V

S→ Sl S	U→ U U	V→ V V V
S→ λ	U→ λ	V→ λ
Sl→ Pair	U→ Pair	V→ Pair

Figure 2: *The composite grammar (S2) for the simple explanations. The only semantic restriction is that* Pair *must map to 1 letter and 1 phoneme.*

X→F	F→ Pair X
X→FL	FL→ Pair X Pair
X→VOLS	VOLS→ X VOL VOL X
	VOL→ Pair
X→KD	KD→ X mapping X

Figure 3: *The additions to the composite simple grammar for the knowledgeable explanations (KD). The semantic restrictions are that a* VOL *must refer to a vowel and that a* mapping *must refer to some rule that is already known and that applies to this training pair.*

2.5 Mob Details

2.5.1 Domain Details

The words used with MOB were the same words that were used in Lehnert's PRO system [1987]. These are approximately 1,500 English words of 3 to 7 letters in length. Training is done with with short words first and other words presented in order of increasing word length. There are 5 sets of 25 test words, one each for each word length. These test words are never used for training. The words were arbitrarily selected from the New World Dictionary [Guralnik, 1980].

2.6 Evaluation of Generalizations

The EBL theory used here can't distinguish between the different generalizations. MOB produces generalizations only with support from the theory and all generalizations have equal support. Therefore the theory may produce erroneous explanations, in that the set of rules it proposes may not be the ones that provide the best generalization to novel words. One phase of MOB will test the utility of the generated generalizations. Utility is tested quite simply: the performance system is run with the addition of the suggested rule. If the inclusion of this rule increases MOB's performance on this word[1] then MOB accepts the generalization. If there is no such improvement then the generalization is rejected. The rules are applied by a simple, forward chaining, production system. The representation used for rules is straightforward. Both sides of the production are literal representations of either the letters or phonemes in a generalization.

After a training word is successfully pronounced the generalizations added

[1] Better performance is defined as increasing the number of correct phonemes in the correct order.

during this training instance are reviewed. Any generalizations that helped in some intermediate stage but weren't used in the final correct pronunciation are deleted.

3 Evaluating MOB

MOB is evaluated on the basis of its ability to generalize and produce the correct pronunciation for words it hasn't been trained on. As is common in this domain performance will be scored on the percent of phonemes correctly generated, not on the number of completely correct words [Lehnert, 1987] [Stanfill and Waltz, 1986] [Sejnowski and Rosenberg, 1988].

In particular MOB is repeatedly tested on its performance on 25 unseen words of each word length. The testing takes place after every 200 phonemes of training words, and just before words of a new length are introduced.

To demonstrate the effects of explanations on search MOB uses 2 kinds of explanations are used, one simple (called S2), one more knowledge based (KD). The simple one will not constrain search much and therefore lead to poor performance. The knowledge based one will give explanations that are constrained by prior learning and therefore give a more restricted search space.

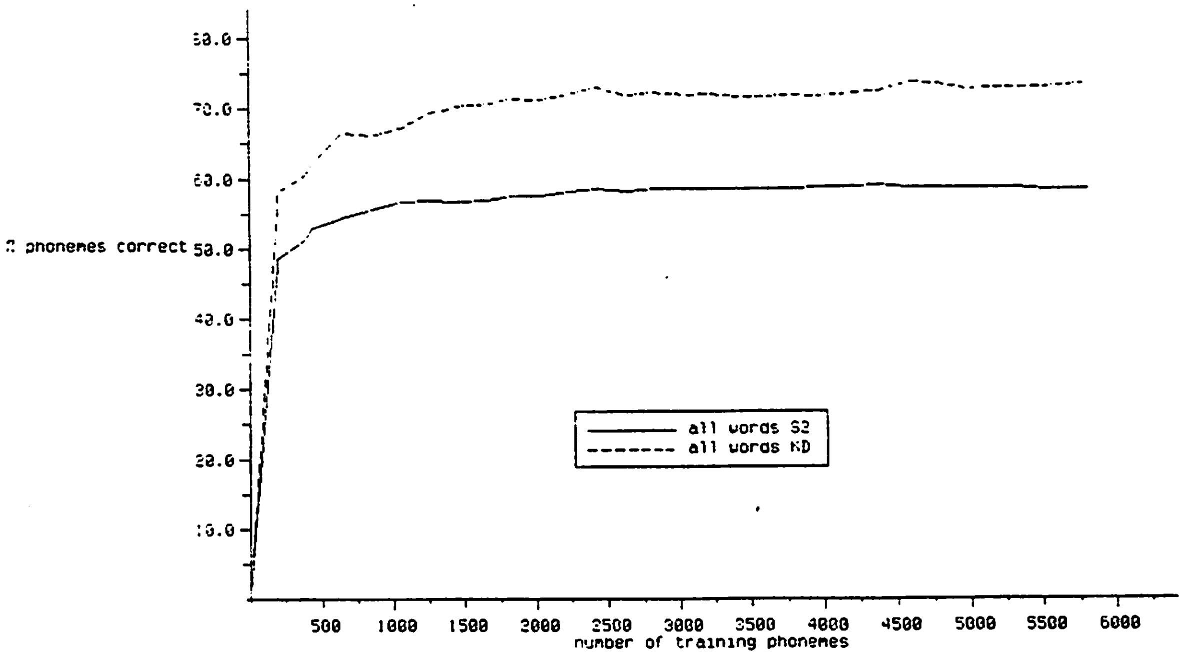
3.1 Results of MOB

3.1.1 MOB as a Learning System

Figure 4 shows the results of training MOB with two different explanation types. The system shows a dramatic increase in performance, this increase is stable and it is far above a chance level of performance. For MOB I have defined chance to be choosing an output phoneme randomly out of all the possible phonemes. This is $\frac{1}{51}$ or approximately 2%. Note that the explanation techniques have a large effect on the performance of the system. The one that uses the more constrained explanations gives better performance.

3.1.2 Comparison to Other Systems

A number of other systems have dealt with this learning problem. Figure 5 shows their performance along with MOB's. MOB doesn't give the best performance, but it has quite respectable performance compared to these systems.



percent correct pronunciation as a function of training phonemes

Figure 4: *Average performance for the S2 and KD explanations*

SYSTEM	% correct on unfamiliar	words of training	training word length
MOB	72	1439	3,4,5,6,7
"	81	"	4,5
MBRtalk[a]	86	4438	?
PRO[b]	75	750	4,5
NETtalk[c]	78	1024	?
chance	2		

[a] [Stanfill and Waltz, 1986]
[b] [Lehnert, 1987]
[c] [Sejnowski and Rosenberg, 1988]

Figure 5: *Comparison of overall percent correct on unfamiliar words for several pronunciation systems*

4 Conclusion

4.1 What MOB Implies

MOB establishes that an EBL technique is useful even when a good domain theory doesn't exist. It gets good performance (comparable to other systems in this domain) even though proof is impossible in this domain.

Since this paper argues that the utility of an explanation is in the restrictions that it places on generalizations, and not the absolute truth or falsity of the inferences we can now look at a large number of possible explanation and generalization methods. Context regression is a single generalization method, likely to be useful in the situation when the appropriate context is represented openly in the training examples.

Open context relies on the existence of fragments of the input that can be translated into the correct output based on their local context. This is most likely the case with problems where the task is some sort of structural transformation. Many tasks are not like this. Problems, such as classification, that have complex inputs but just a few simple acceptable answers are not suitable. In these problems there is no mapping of substructure to substructure. Without meaningful substructures context regression is unlikely to find generalizations that transfer from one example to another.

4.2 Future Directions

4.2.1 Different inference techniques

There are two different places where MOB uses inference. One is in constructing proofs. The other is inferring generalizations from proofs. The use of explanation trees blurs the distinction in MOB, but there is no reason why these two methods need to be the same. For instance, when constructing a proof, one might simply assume some truth values that aren't really known. The generalization technique might know nothing about this and so do a straightforward regression assuming that the proof was a normal FOL proof. This leaves room for exploring techniques for plausible explanations and not requiring altering the generalization method. On the other hand there are techniques such as context regression which don't require FOL as a basis for explanation or generalization.

4.2.2 Hybrid Techniques

MOB suggests a method for dealing with partially learned theory. Use the knowledge that is available and discriminate between multiple explanations via testing. If the testing suggests further theory refinement then do so, otherwise testing will provide reasonable generalizations. The combination of the imperfect theory tolerance of MOB and the theory enhancement techniques of other researchers may give a system the capacity to function before it can understand a domain well, but also use better information as it can be found.

References

[Clancey, 1988] William J. Clancey. Detecting and coping with failure. In *Proceedings of 1988 Spring Symposium Series, Explanation-Based Learning*, pages 22–25, 1988.

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

[Guralnik, 1980] David B. Guralnik, editor. *New World Dictionary, Second College Edition.* William Collins Publishers Inc., 1980.

[Lehnert, 1987] Wendy Lehnert. Cased-based problem solving with a large knowledge base of learned cases. In *Proceedings of AAAI-87*, pages 301–306, 1987.

[Mitchell *et al.*, 1986] Tom. M. Mitchell, Richard M. Keller, and Smadar T. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1:47–80, 1986.

[Rajamoney, 1988] Shankar Rajamoney. Expermentation-based theory revision. In *Proceedings of 1988 Spring Symposium Series, Explanation-Based Learning*, pages 7–11, 1988.

[Sejnowski and Rosenberg, 1988] T. Sejnowski and C. Rosenberg. Nettalk: a parallel network that learns to read aloud. In J. A. Anderson and E. Rosenfeld, editors, *Neurocomputing Foundations of Research*, chapter 40, pages 661–672. MIT Press, 1988. also Johns Hopkins University Electrical Engineering and Computer Science Technical Report JHU/EECS-86/01.

[Stanfill and Waltz, 1986] C. Stanfill and D. Waltz. Towards memory-based reasoning. *Communications of the ACM*, 12(12):1213–1228, December 1986.

[VanLehn, 1987] Kurt VanLehn. Learning one subprocedure per lesson. *Artificial Intelligence*, 31(1):1–40, 1987.