Title
Efficient Learning of Language Categories: The Closed-Category Relevance Property and Auxiliary Verbs

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 12(0)

Authors
Nicholl, Sheldon
Wilkins, David C.

Publication Date
1990

Peer reviewed

Efficient Learning of Language Categories: The Closed-Category Relevance Property and Auxiliary Verbs

Sheldon Nicholl and David C. Wilkins

nicholl@cs.uiuc.edu
wilkins@cs.uiuc.edu
Department of Computer Science
University of Illinois
405 North Mathews Avenue
Urbana, IL 61801

Abstract

This paper describes the mechanism used by the ALACK language acquisition program for identification of auxiliary verbs. Pinker's approach to this problem (Pinker, 1984) is a general learning algorithm that can learn any Boolean function but takes time exponential in the number of feature dimensions. In this paper, we describe an approach that improves upon Pinker's method by introducting the Closed-Category Relevance Property, and showing how it provides the basis of an algorithm that learns the class of Boolean functions that is believed sufficient for natural language, and does not require more than linear time as feature dimensions are added.

1 Introduction

Within the study of language acquisition, the problem of category identification is still a challenge to formal theories of language acquisition. Even the identification of the members of a closed category such as the Auxiliary Verbs (hereinafter referred to as AUX) stands unresolved. The principal approaches to this problem include (Anderson, 1983), (Berwick, 1985), and (Pinker, 1984); Our approach is closest to that of Pinker. We share with Pinker the following five assumptions: (1) Language is learned not from a string of words alone but from the corresponding meaning (and possibly other attributes) as well. (2) Some components of the meaning can be represented with features. (3) The features are drawn from sets we will call *feature dimensions*. Examples of several feature dimensions are shown below in Table 1. (4) Candidate AUXes are not annotated with syntactic features in the input, e.g. (Berwick, 1985), nor is the input prechunked into phrase-like groupings, e.g., (Anderson, 1983). (5) Steele's cross-linguistic generalization (Steele et al., 1981), holds for AUXes: AUXes encode tense or modality or both.

Pinker's approach is based on a general learning algorithm that (i) can learn any Boolean function but (ii) takes time exponential in the number of feature dimensions. In contrast, our approach, which depends on the Closed-Category Relevance Property, defined Section 2, (i) cannot learn an arbitrary Boolean function but is conjectured to be sufficient for natural language, and (ii) does not require more than linear time as feature dimensions are added.

2 The Closed-Category Relevance Property

In this section, we introduce the Closed-Category Relevance Property that will be used to identify the category AUX; this is the basis of our algorithm for AUX identification. Our goal is identification only, not full control. So to recognize that **are**, for example, is an auxiliary is sufficient; to learn the rules that control when to choose **are** instead of **is** or **were**, for example, is beyond the scope of this paper.

The Closed-Category Relevance Property can be viewed abstractly as a two-place predicate R which takes a word as its first argument and a feature dimension as its second argument: $R(word, dimension)$. Its purpose is to relate words to features that control their usage. It is meant to capture the idea that *animacy*, for example, has no bearing on the word **are**, i.e., $\neg R(\textbf{are}, animacy)$, but that *tense*, for instance, does: $R(\textbf{are}, tense)$.

The Closed-Category Relevance Property is defined as follows: $R(word, dimension)$ is true **iff** *word* can encode some but not all of the values in *dimension*. $R(\textbf{are}, tense)$ is true because **are** can encode present and future time but not past time. The definition is satisfied because **are** can encode some but not all of the values on the tense dimension. $R(\textbf{are}, animacy)$ is false bacause **are** can be used in both animate ("The dogs are running") and inanimate ("The computers are running") contexts, so it trivially encodes both the values on the animacy dimension.

3 The AUX Learning Algorithm

The Closed-Category Relevance Property has been embodied in a computer model of language acquisition called ALACK, which runs in Common Lisp on a Sun 4.

3.1 The Input to the Algorithm

The input to ALACK relevant to this discussion consists of: (1) a segmented string of words, each of which is segmented into grammatical morphemes, (2) a list of semantic categories corresponding to each word, drawn from the set {thing,event,state,null} where thing marks a perceptually salient physical object, event marks a perceptually salient event, action, or process, state marks some ongoing state, and null is the default which applies to all other words, and (3) a list of sets of feature values, which are empty sets for words marked null, nonempty otherwise.

3.2 The Algorithm and an Example

In this section, we describe the steps of our Closed-Category Relevance algorithm, and show how each step processes the following input that is based on actual input to ALACK:

person	*1st-person, 2nd-person, 3rd-person*
number	*singular, plural*
tense	*past, present, future*
animacy	*animate, inanimate*
modality	*modal, nonmodal*
aspect	*perfective, imperfective, progressive*

Table 1: Features used as examples throughout the text. They, and the actual values within them, are used as illustration in the text, and of course do not constitute a claim as to what is indeed linguistically complete and correct.

```
(((the) (men) (are) (walk ing))
 (null thing null event)
 (() (3rd-person plural animate) () (present progressive)))
```

The feature values are drawn from the same sets of *feature dimensions* described in the introduction. ALACK embodies the Closed-Category Relevance Property in a manner logically equivalent to the following.

- Step 1. All words marked **null** are collected into a set E. In this case, $E = \{\textbf{are}, \textbf{the}\}$. It is this set that is tested for closed-category relevance; the treatment of the other words is beyond the scope of this paper.
- Step 2. All the feature values are collected into a set F. In this case, $F = \{$*3rd-person, plural, animate, present, progressive*$\}$. ALACK filters the input to make sure that F contains only one value from each feature dimension. Inputs violating this constraint are ignored.
- Step 3. Unless they have been built already, ALACK constructs all the triples $E \times G \times H$ where

$$G = \{x \mid x \subseteq F \wedge |x| = 1\}$$

$$H = \{dim(f) \mid f \in F\}$$

$$dim(f) = y \quad \text{if } f \in y$$

  For example, $dim(singular) = number$, since $number = \{singular, plural\}$. Three examples of $E \times G \times H$ are $(are, \{present\}, tense)$, $(are, \{animate\}, animacy)$, and $(the, \{plural\}, number)$.

- Step 4. If a triple (e, S, d) where $e \in E$ and $d \in dim(F)$ has already been constructed, set S is updated to include the new feature value $f \in F$: $S_{new} = S_{old} \cup \{f\}$. So for example, if ALACK gets the following input

```
(((the) (ball) (is) (fall ing))
 (null thing null event)
 (() (3rd-person singular inanimate) () (present progressive)))
```

  then the example triple $(are, \{animate\}, animacy)$ is updated to

$$(are, \{animate, inanimate\}, animacy)$$

- Step 5. All the updated triples are tested against the following rule:

$$(x, y, y) \Rightarrow \neg R(x, y)$$

  Since $animacy = \{animate, inanimate\}$, the example triple matches the left-hand side, which forces the conclusion $\neg R(\textbf{are}, animacy)$; i.e., **are** is not closed-category relevant for animacy, as discussed above. But are is still relevant for *tense*: the triple $(are, \{present\}, tense)$ has not been changed. Pinker's model and ours agree that all dimensions not explicitly found to be irrelevant are relevant by default.

ALACK organizes some feature dimensions into a set, or *domain*, which we will call the *verbal* domain. In the current implementation of ALACK, this domain is as follows:

$$verbal = \{tense,\ aspect\}$$

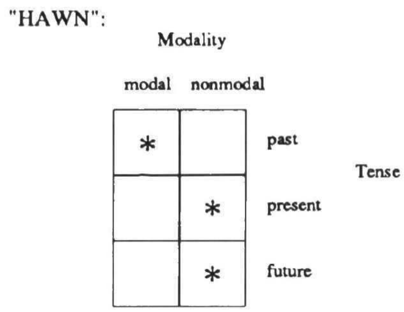
Formally, category inference is done as follows. Given morpheme x and dimension y,

$$R(x, y) \wedge (y \in verbal) \Rightarrow x \in AUX$$

The rule is justified by the previous discussion on Steele's generalization (Steele et al., 1981). ALACK's implementation does not yet include modality; aspect has been included since it works well for English. For example, since $R(\mathbf{are}, tense)$ and $tense \in verbal$, the rule allows $\mathbf{are} \in AUX$ to be concluded.

4 Analysis of AUX Learning Algorithm

Correctness. Clearly the closed-category relevance property leads to an algorithm that is not correct for all Boolean functions. For example, suppose that English had an auxiliary **hawn** whose paradigm, or grammar chart, looked like this:



where the stars mean to use the auxiliary, and the blanks mean ϕ (the null morpheme), for instance. The auxiliary **hawn** is irrelevant along both the *tense* and *modality* dimensions, even though both dimensions are (1) important to auxiliary identification, as discussed below, and (2) important to the proper usage of **hawn**. Though Closed-Category Relevance is not correct from the standpoint of full logical generality, it is only a small stipulation beyond the generalization of (Steele 1981); English has nothing like **hawn**. This leads to the **Closed-Category Relevance Conjecture:** All the auxiliaries in all the world's languages that have auxiliaries are relevant for tense or modality or both. This Conjecture goes beyond Steele's generalization in that the Conjecture would not allow something like **hawn**, while Steele's generalization would. Steele's generalization is satisfied by any encoding of tense or modality; the Conjecture demands that the encoding satisfy the Relevance Property as well.

Complexity Analysis. We wish to examine the time complexity of determining $\neg R(w, d)$ for a given w, d as the total number of dimensions T in *features* is increased. For the sake of this argument, we can stipulate that in the implementation, the triples (e, S, d) are accessed by their dimension via a function h, where $h(d) = \{(e, S, d') \mid d' = d\}$. If h is chosen to be implemented in an array A, one may simply search A linearly for the desired dimension d, i.e., the search time is $O(T)$. If A is sorted, the time drops to $O(\log(T))$. If h is a hash function, that time can be lowered significantly.

5 Cross-Linguistic Analysis of Relevance

The idea of Closed-Category Relevance has interesting implications when applied to Bickerton's work (Bickerton, 1984). Suppose we are given the problem of identifying the auxiliaries in some of the creole languages that Bickerton has studied. For instance, in Hawaiian Creole, three auxiliaries to try are **bin**, **go**, and **stei**. For Lesser Antilles Creole, the auxiliaries are **ka**, **ke**, and **te**. Two of the auxiliaries in Saramaccan are **ta**, and **bi-o-ta**. It is possible to translate his notation for tense, modality, and aspect (see Table 1, p. 183) into this system in the following way. Let $features = verbal = \{tense, modality, aspect\}$ where $tense = \{anterior, nonanterior\}$, $modality = \{realis, irrealis\}$, and $aspect = \{punctual, nonpunctual\}$. Then for Hawaiian creole, *R(***stei***, aspect)*, *R(***go***, modality)*, *R(***bin***,tense)*, and for Lesser Antilles Creole, *R(***ka***,aspect)*, *R(***ke***,modality)*, and *R(***te***,tense)*. Now in a case like Saramaccan where auxiliaries are built up morphologically, closed-category relevance can be applied successfully to each morpheme individually or to a whole word, e.g. *R(***bi-o-ta***,tense)* or *R(***ta***, aspect)*. Closed-Category Relevance is confirmed for auxiliaries in these languages. Although the above demonstration hardly constitutes a full confirmation of Closed-Category Relevance, it does show that further tests of Relevance in other languages are worthwhile. It also lends credence to the idea of Closed-Category Relevance as an acquisition principle.

6 Comparison to Pinker's Learning Method

This section will do three things: show that Pinker's Method is logically correct, analyze its computational complexity, and compare it to Closed-Category Relevance.

6.1 Correctness

The set of features described in Table 1 can be viewed as an instance of the following format, where the given set of features is simply a set of dimensions, and each dimension is simply a set of values. To make the statement of Pinker's method more precise, we introduce here a set of *feature-names*, with the obvious bijection between *features* and *feature-names*: $features = \{dim_1, dim_2, \ldots, dim_n\}$, $dim_i = \{val_1, val_2, \ldots, val_{f(i)}\}$, and $feature\text{-}names = \{d_1, d_2, \ldots, d_n\}$. The input to Pinker's procedure is first a sequence of *attribute-lists*, which we will index with $q \in \{1, 2, 3, \ldots\}$. Each attribute-list is a set of pairs, with a feature name in the first slot of the pair and a value from the corresponding feature dimension in the second:

$$attribute\text{-}list_q = \{(d_{q_1}, val_{q_1}), (d_{q_2}, val_{q_2}), \ldots, (d_{q_n}, val_{q_n})\}$$

The procedure is also provided with a morpheme m_i on each trial, where

$$m_{i_q} \in M = \{m_1, m_2, \ldots, m_i, \ldots, m_{g(q)}\}$$

The goal is to discover (learn) the Boolean expression B_j for each m_i, where B_j is a possibly complex Boolean expression built up from the pairs in the *attribute-lists*, using conjunction and disjunction only, without negation, and $m_i \Leftrightarrow B_j$.

Pinker's solution to this learning problem is to build a big multi-dimensional array, a *Paradigm*, and to fill single array locations by (1) reading each input q as a set of coordinates and (2) placing m_{i_q} at the location specified by these coordinates. The array is built up one dimension at a time; the dimensions to add are selected at random from the image of $\{x \mid (x, y) \in attribute\text{-}list_q\}$, for some q. That is, the array can be built up only from dimensions that occur somewhere in the input. Pinker does not say what to do with the morpheme entries when a new dimension is added; one possibility would be simply to forget them all and start over with a bigger matrix. We will neglect this problem and assume that we begin with a big enough matrix.

Given a sufficiently large matrix, Pinker's method is correct, since it is simply storing the examples into the matrix as they come in. Although this result is new (Pinker didn't give correctness proofs), it is very minor, and serves only to provide a background for the next result, which is the complexity result.

6.2 Complexity

Since Pinker's learning procedure P chooses feature dimensions dim_i at random, it is quite possible that P will choose an irrelevant dimension, say dim_5, even though B only requires the use of some other dimensions, say dim_1, dim_2, and dim_3. It is important to dispose of irrelevant dimensions, since, among other things, Pinker's method for finding auxiliaries does not tolerate irrelevant dimensions. We will first give Pinker's statement of his method for disposing of irrelevant dimensions, show its correctness, and then give its time complexity.

Here is Pinker's (Pinker, 1984) method for eliminating irrelevant dimensions, which he calls Procedure I3. His use of "cells" corresponds to "array locations" in this paper; similarly, "paradigm" means "array", and "affix" means "morpheme":

> If the same affix appears in all the cells defining a dimension across a given combination of values of the other dimensions, and this is true for every possible combination of values of the other dimensions, eliminate that entire dimension from the paradigm. (Pinker, 1984) p. 186.

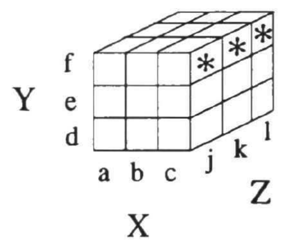
For example, suppose that $features = \{X, Y, Z\}$, $X = \{a, b, c\}$, $Y = \{d, e, f\}$, $Z = \{j, k, l\}$, and $feature\text{-}names = \{\mathbf{X},\mathbf{Y},\mathbf{Z}\}$, and suppose further that the input consists of the following three input attribute-lists for morpheme m:

$$attribute\text{-}list_1 = \{(\mathbf{X}, c), (\mathbf{Y}, f), (\mathbf{Z}, j)\}$$

$$attribute\text{-}list_2 = \{(\mathbf{X}, c), (\mathbf{Y}, f), (\mathbf{Z}, k)\}$$

$$attribute\text{-}list_3 = \{(\mathbf{X}, c), (\mathbf{Y}, f), (\mathbf{Z}, l)\}$$

The (complete) array resulting from these inputs is:



where the asterisks correspond to the three inputs. Pinker's elimination method says that the Z dimension can be eliminated from the array, leaving just dimensions X and Y. We now show that this is logically correct.

Note that the possible attribute-lists for m as shown in the array can be represented as follows:

$$cfj \vee cfk \vee cfl = cf(j \vee k \vee l) = cf(true) = cf$$

since $j \vee k \vee l \Rightarrow true$. Now since the resulting expression, cf, contains no reference to the Z dimension, that dimension can be eliminated. It should now be clear that this argument can be carried out in general for any number of dimensions, values, and inputs.

Procedure I3 can now be stated formally. Let $S = \{f_1, f_2, \ldots, f_k\}$ be a set of feature dimensions. Let $B(S)$ be a Boolean expression based on the members of S. Given m iff $B(S)$, dimension $f_i \in S$ can be eliminated from S only if $[B(S)$ *iff* $B(S - \{f_i\})]$. In our example, $S = \{X, Y, Z\}$ and $f_i = Z$. So

$$B(S) = B(\{X, Y, Z\}) = cfj \vee cfk \vee cfl = cf = B(\{X, Y\}) = B(S - \{f_i\})$$

This is just a formal way of saying that dimension Z can be eliminated from consideration, just as it was in the array realization above.

Result: Procedure I3 is NP-complete. Proof: The elimination of f_i from S requires that the expression $B(S) \iff B(S - \{f_i\})$ be shown to be a tautology. The tautology problem is NP-complete.

This result will be used shortly. First, a predicate similar in spirit to our Closed-Category Relevance predicate can be defined: $R'(word, dimension)$ iff $(word \Leftrightarrow B(S)) \wedge (dimension \in S)$. Pinker's AUX identification method can now be approximated by the following expression. Pinker left several terms mathematically undefined.

$$Prob(R'(word, tense) \vee R'(word, modality), Phon(word), Syn(word)) \Rightarrow word \in AUX$$

That is, a word is an auxiliary if it satisfies an undefined predicate ($Prob$) based on a probabilistic combination of its arguments: the ability of the word to encode tense or modality (R'), an undefined predicate ($Phon$) based on certain phonological properties of the word, and an undefined predicate (Syn) based on certain other syntactic properties of the word. Now if $Prob$ is strict in its first argument, $Prob$ must be at least NP-complete. Hence Pinker's auxiliary identification procedure is at least NP-complete under the assumption that $Prob$ needs the output of R'.

6.3 Comparison to Closed-Category Relevance

Figure 1 showed that there exist logical, if not linguistic, counterexamples to Closed-Category Relevance. By contrast, section 6.1 showed that under assumptions (1) – (5) in section 1, Pinker's procedure could handle any logically possible AUX rule, including Figure 1. Section 4 showed, however, that Closed-Category Relevance leads to an algorithm that is fast, while section 6.2 showed that Pinker's procedure is NP-complete.

7 Conclusion

The Closed-Category Relevance Property has been defined and has been shown to lead to an efficient algorithm for the identification of Auxiliary Verbs. Relevance was shown to hold in several languages other than English, demonstrating that the same algorithm could be applied to Auxiliary Verb identification in those languages. Finally, Pinker's method was subjected to an analysis which, under certain reasonable assumptions, proved it logically correct but NP-complete.

Acknowledgements

The authors would like to thank Cindy Fisher, Peter Haddawy, and Steve Pinker for their very helpful comments.

References

Anderson, J. R. (1983). *The Architecture of Cognition.* Cambridge, MA: Harvard.

Berwick, R. C. (1985). *The Acquisition of Syntactic Knowledge.* Cambridge, MA: MIT Press.

Bickerton, D. (1984). The language bioprogram hypothesis. *The Behavioral and Brain Sciences,* 7:173–221.

Pinker, S. (1984). *Language Learnability and Language Development.* Cambridge, MA: Harvard.

Steele, S., Akmajian, A., Demers, R., Jelinek, E., Kitagawa, C., Oehrle, R., and Wasow, T. (1981). *An Encyclopedia of AUX: a study of cross-linguistic equivalence.* Cambridge, MA: MIT Press.