COGNITIVE SCIENCE 6, 277-290 (1982)

Ideal Learning Machines*

DANIEL N. OSHERSON
Center for Cognitive Science
M.I.T.
Cambridge, MA

MICHAEL STOB
Department of Mathematics
Calvin College

SCOTT WEINSTEIN
Department of Philosophy
University of Pennsylvania

We examine the prospects for finding "best possible" or "ideal" computing machines for various learning tasks. For this purpose, several precise senses of "ideal machine" are considered within the context of formal learning theory. Generally negative results are provided concerning the existence of ideal learning-machines in the senses considered.

SECTION 1: INTRODUCTION

Machines that learn have both technological and scientific interest. On the one hand, they hold the promise of uncovering useful regularities that would otherwise be missed. On the other hand, they serve as potential models of human learning, including language acquisition.[1] Both endeavors have met with such initial success that it is natural to inquire about ideal or "best possible" learning-machines for a given problem domain.[2] Such an inquiry requires clarification of the sense in which a machine might be ideal;

*The second author was partially supported by NSF Grant MCS 80-02937.

[1]For a general approach to machine learning within the Artificial Intelligence tradition, see Winston (1978). Wexler and Culicover (1980, Chs. 1, 2) provide a valuable introduction to the empirical issues confronting the application of formal learning theory to human language acquisition.

[2]All the learning-machines to be considered in this paper will be "idealizations" of physically real learners, in that factors like longevity will be left out of account. At issue are ideal versions of such idealized learners, that is, formal devices with superior inductive powers in one or another sense. In what follows, we use the term "ideal" exclusively with the latter meaning.

more than one clarification suggests itself. Thus, a machine might be considered ideal with respect to how much it can learn; for example, a machine that could learn every possible language presented to it would be ideal in a strong sense. Alternatively, it might be the manner in which a machine learns that is ideal; for example, an ideal machine for a given class of languages might learn those languages at least as fast as any other machine.

Ideal learning-machines, in one or another sense, have evident technological importance. They are of interest as well to those who wonder whether the human infant is an ideal learner of natural language, or a close approximation thereto; such an evolutionary development might have imposed strong constraints on the class of possible human grammars.

The purpose of the present paper is to evaluate in general terms the prospects for finding ideal learning-machines of various sorts. For this purpose it will be necessary to precisely render *learning* and related notions. From among the several options available, we rely on two definitions due to Gold (1967); these characterizations figure prominently in contemporary work in formal learning theory. In the context of Gold's definitions it is possible to distinguish four precise senses of "ideal learning-machine." For each sense we ask whether such learning-machines exist.

In more detail, we proceed as follows. Section 2 presents the formal paradigm of learning that is most relevant to current theories of language acquisition; this paradigm is called "text-identification." Section 3 considers four construals of ideal text-identification, and provides mainly negative results about the existence of machines with such ideal capacities. Section 4 introduces a different learning paradigm, called "informant-identification." The subsequent section studies four construals of ideal informant-identification parallel to those given in Section 3; different, but still mainly negative results are provided. Section 6 includes a brief summary.

To conform to standard usage, learning machines will be said to respond to "languages." Languages are construed extensionally, i.e., as sets of sentences, but they can also be understood more generally, as sets of data of any nature (so long as each datum is a finite object). In fact, it is sufficient for present purposes to restrict attention to numerical languages, i.e., to subsets of the set, N, of natural numbers. Nonnumerical data can be assimilated to the present discussion by means of coding techniques.[3] Likewise, "grammars" for languages will be construed broadly, as certain kinds of tests for membership in subsets of N.

SECTION 2: TEXT-IDENTIFICATION

In this section we present Gold's (1967) definition of "identification in the limit by arbitrary text," to be abbreviated to "text-identification."

[3] For a discussion of coding, see Boolos and Jeffrey (1980). The latter serves, as well, as an introduction to Turing Machines, Church's Thesis, and related matters mentioned below.

2.1 Languages and Positive Tests

We identify learning-machines with ordinary Turing Machines, relying on Church's Thesis to insure that every intuitively mechanical process is adequately represented thereby. Each Turing Machine is conceived as examining input languages (i.e., sets of numbers) and announcing output grammars.

Grammars are identified with *positive tests.* A positive test is a procedure, P, such that for some language, L, P behaves as follows for all $x \in N$: given x as input, P eventually halts with output 1 if and only if $x \in L$.[4] In this case, P is said to be *for* L. It is an elementary result of automata theory that if there is one positive test (i.e., Turing Machine) for a given language, then there are an infinity of distinct positive tests for that language. Such Turing Machines provide a convenient normal form for positive tests, learning-machines can be conceived as announcing the programs of other Turing Machines; but we shall continue to refer to the latter as positive tests.

Henceforth we restrict attention to languages for which there are positive tests. Such languages are called *recursively enumerable.* There are recursively enumerable languages without recursively enumerable complements; one such language (to play a role in our discussion) is standardly denoted: K.

2.2 Texts

Languages are presented in piecemeal fashion to learning-machines; such presentations are called *texts.* More precisely, let L be a language. A *text for* L is any infinite sequence of members of L, repetitions allowed, such that every member of L occurs at least once in the sequence.

A finite sequence that constitutes an initial segment of a text, t, is said to be *in* t. If s is a finite sequence, *Rng(s)* denotes the (finite) set of elements occurring in s. We let *SEQ* stand for the set of all finite sequences.

To illustrate, let L = {s1, s2, s3, . . . } be a language. A sample text, t, for L begins as follows: s3, s2, s2, s3, s1, s2, ... The first five finite sequences in t are:

(*)

s3
s3, s2
s3, s2, s2
s3, s2, s2, s3
s3, s2, s2, s3, s1

These finite sequences are said to have *lengths* one through five, respectively. In contrast, s2, s2, s3 is not a finite sequence in t since it is not initial in t. The text t is not a sequence in itself, since t is not finite.

[4]If $x \notin L$, P may halt with some output other than 1, or P may never halt at all; the definition of positive test is not specific in this regard. P is a "positive" test for L in the sense that it is guaranteed to react to membership in L (positive tests are not guaranteed to react to nonmembership).

2.3 Convergence and Text-Identification

Each learning-machine, M, constitutes a possibly partial function from SEQ into the set of all positive tests (i.e., into the set of Turing Machines). The function defined by a given machine, M, may be partial because M may yield no output in response to some members of SEQ.

Given a text, t, we may conceive of M as examining ever longer finite sequences in t, as t is fed one element at a time into M. For some or all of these finite sequences, M announces positive tests. Each time M announces a positive test, the next element of t (and hence the next finite sequence in t) is fed into M; if M is undefined on a certain finite sequence, s, in t, and thus never announces a positive test in response to s, no more of t is fed into M. Given $s \epsilon SEQ$, *M(s)* denotes the positive test, if any, produced by M after it has read all of s (if M is defined on each sequence in s).

Given a learning-machine, M, a text, t, and a positive test, p, we say that M *converges to* p *on* t just in case (a) M yields a positive test on every finite sequence in t, and (b) there is an n such that for all finite sequences, s, in t such that the length of s exceeds n, M applied to s yields p. Intuitively, M converges to p on t in case M is defined on all initial segments of t, and M's output eventually stabilizes to p. If M fails to converge to any positive test on a given text, t, then M is said to *diverge* on t. M is said to *text-identify* a language, L, just in case for every text, t, for L there is a positive test, p, for L such that M converges to p on t.[5]

Let *L* be a class of languages. M is said to *text-identify L* just in case M text-identifies every language in *L*. A class of languages that some learning-machine text-identifies is called *text-identifiable*. Notice that if M text-identifies a collection, *L*, of languages, then M text-identifies every subset of *L* as well (hence, every subset of a text identifiable class of languages is also text-identifiable). Finally, note that every (recursively enumerable) language, L, taken by itself, is text-identifiable; for example, a machine that always conjectures the same positive test for L, regardless of input, text-identifies L. In contrast, questions about the text-identifiability of collections of languages are often nontrivial, since many such questions receive negative answers (as will be seen directly). Such is the consequence of requiring the same learning machine to respond correctly to texts for different languages.

The set of machines that text-identifies a given class, *L*, of languages is denoted: M^t_L.

[5]There are thus three ways that M can fail to text-identify a language, L: for some text, t, for L (a) M might converge to a positive test that is not for L, (b) M might be undefined on some finite sequence in t (and thus diverge on t), or (c) M might forever offer distinct positive tests in response to finite sequences in t (and thus likewise diverge, whether or not some or all of those positive tests are for L).

SECTION 3: IDEAL TEXT-IDENTIFICATION

In this section we specify four senses of "ideal learning-machine for text-identification." Relevant existence or nonexistence theorems are provided. All definitions are relative to text-identification.

3.1 Complete Text-Identification

A learning-machine, M, that can text-identify every (r.e.) language would be ideal in an obvious sense. Such machines are called *complete.*[6] The following proposition shows that there are no complete learning-machines.

Proposition 1. Let F^* be any collection of languages consisting of all finite subsets of N, plus any infinite subset of N. Then F^* is not text-identifiable by any learning-machine.

Proof. The proof of the proposition may be found in Gold (1967). □

Proposition 1 shows that our conception of "ideal learning-machine" must be weakened if it is to be nonvacuous. One's initial impulse is to restrict attention to text-identifiable collections of languages, and to define a machine, M, to be ideal just in case it text-identifies any such class. However, it is easy to see that no ideal machine of this kind exists. For, every singleton collection of languages, {L}, is text-identifiable, and Proposition 1 is enough to show that no machine can text-identify every such collection. To frame a nonvacuous definition of ideal learning machine it is necessary to impose additional limitations. The next subsection presents a plausible way of doing this. We shall see that despite our maneuvers the resulting definition does not escape triviality.

3.2 Maximal Text-Identification

Since the proof of Proposition 1 turns on a collection of languages that is not text-identifiable, we are led to the following definition. A learning-machine, M, is said to be *maximal* just in case M text-identifies a collection, L, of languages such that no proper superset of L is text-identifiable; in this case, M is said to be maximal *for L*. The learning ability of a maximal learning-machine is therefore unsurpassed by any other learning machine. Such machines are "ideal" for L in a strong sense.

[6]Or rather: *complete (for text-identification)*. We suppress the parenthetical qualification henceforth in this section.

There are maximal machines for the collection of all finite languages. To see this, consider a machine, M, that behaves as follows: Given any finite sequence, s, M produces a positive test for exactly Rng(s); such a positive test can be uniformly and mechanically constructed from a given finite set. It is clear that if M is supplied with a text for a finite language, it will converge to a positive test for that language; and if M is supplied with a text for an infinite language, M will never converge on that text. Hence, M text-identifies the collection, F, of all finite languages and nothing else. By Proposition 1, no proper superset of F is text-identifiable; hence, M is maximal for the set of all finite languages.

Are there maximal machines for collections other than the collection of all finite languages? The next Proposition shows that this is not the case.

Proposition 2. Let L be a collection of languages such that L is text-identifiable, and no proper superset of L is text-identifiable. Then L is the class, F, of all finite languages.

Proof. On the one hand, suppose that L includes no infinite language. Then, if L is a proper subset of F, there is a proper superset of L (namely, F) that is text-identifiable; and if L is exactly F, then Proposition 1 shows that no proper superset of L is text-identifiable.

On the other hand, suppose that L includes an infinite language, L. We show that there is another infinite language, L′, such that $L \cup \{L'\}$ is text-identifiable (and hence L is not maximal). For this purpose, a lemma is useful.

Lemma. Let M be a learning-machine, and let L be a collection of languages that M text-identifies. For every $L \in L$ there is a finite sequence, s, such that $\mathrm{Rng}(s) \subseteq L$, M(s) is a positive test for L, and for every finite extension, s′, of s, such that $\mathrm{Rng}(s') \subseteq L$, $M(s) = M(s')$.

The lemma is proved in Blum and Blum (1975).

Now let M text-identify L. Let s be the finite sequence for L and M specified by the lemma. Let $p \notin L - \mathrm{rng}(s)$. Let $L' = L - \{p\}$. $L' \notin L$ since M must give a positive test for L when given a text which begins with s and ends with the rest of $L - \{p\}$. But $L \cup \{L'\}$ is identifiable by a learning-machine, M′, defined as follows. Given any positive test, k, we let W_k be the language for which k is a positive test. Then, for all $t \in \mathrm{SEQ}$:

$$M'(t) = \begin{cases} \text{a positive test for } W_{M(t)} - \{p\} \text{ if } p \notin \mathrm{rng}(t) \\ M(t), \text{ otherwise.} \end{cases}$$

It is clear that we can mechanically test which case we are in, and mechanically produce a positive test of the sort desired in the first case.[7] □

Proposition 2 shows that with the trivial exception of machines that are restricted to learning finite languages, for any learning machine, M, there is another learning machine, M′, such that M′ text-identifies everything that M text-identifies, plus more. Consequently, if a machine is considered to be ideal only if no other machine text-identifies more than it does, then there are no ideal learning machines that text-identify an infinite language.

3.3 Efficient Text-Identification

The conceptions of "ideal machine" considered so far have concerned what a machine can learn. In this subsection and the next we consider instead how the learning proceeds. As before, it will be necessary to relativize our conception of ideal to particular classes of languages.

If a learning-machine, M, text-identifies a language, L, then for any text, t, for L there is some finite sequence, s, in t such that M begins to converge on t upon examining s. The length of s is called the *convergence length for* M *on* t. In general, different texts for L will be associated with different convergence lengths for M.

A learning-machine, M, is "efficient" with respect to the languages it text-identifies to the extent that M's convergence lengths are small.[8] This suggests another sense of "ideal learning-machine for a given class of languages." Let L be a class of languages, and let M, $M' \epsilon M_L^t$. We say that M is *at least as efficient as* M′ *for L* just in case for all $L \epsilon L$, and for all texts, t, and L, the convergence length for M on t is at least as small as the convergence length for M′ on t. If for all $M' \epsilon M_L^t$, M is at least as efficient as M′ for L, then M is said to be *most efficient for L*.

A weaker notion of ideal efficiency may also be defined. Let L be a class of languages, and let $M \epsilon M_L^t$. M is said to be *maximally efficient for L* just in case for all $M' \epsilon M_L^t$, if M′ is at least as efficient as M for L, then M is at least as efficient as M′ for L.[9]

Obviously, most efficiency implies maximal efficiency; the converse implication is false.[10] Furthermore, there are text-identifiable classes of lan-

[7]An easy adaptation of this proof shows that every text-identifiable class of languages that includes an infinite language can be extended to a text-identifiable collection that includes an infinity of new languages. Similar remarks apply to Proposition 8.

[8]Note that efficient machines need not be "fast" since they may consume considerable time on each input.

[9]Thus, M is maximally efficient for L just in case no machine is both (i) at least as efficient as M for L, and (ii) more efficient than M on some text for a language in L.

[10]For a counterexample, let $L = \{ \{2, 3\}, \{2, 4\} \}$, and let $M \epsilon M_L^t$ conjecture $\{2, 3\}$ when it encounters a 3 (and thereafter), or an initial string of 2's, and let M conjecture $\{2, 4\}$ otherwise. M is maximally efficient for L, but not most efficient; for, M is not at least as efficient as $M' \epsilon M_L^t$ which conjectures $\{2, 4\}$ when it encounters a 4 (and thereafter), or an initial string of 2's, and which conjectures $\{2, 3\}$ otherwise.

guages for which most efficient learning-machines exist; classes containing just one language provide trivial examples, but infinitary cases can also be devised. Can most efficient or maximally efficient machines always be found for text-identifiable classes of languages? The next proposition answers this question negatively.

Proposition 3. There are text-identifiable collections of languages for which there are no maximally efficient learning machines.

Proof. One such collection, L, consists of languages of form

$$L_a = \{2^i, 3^a: i \in N \text{ if } a \in \overline{K}, \text{ and } i \in N - \{a\} \text{ if } a \in K\}$$

for all $a \in N$.

Lemma. For no learning-machine, $M \in M_L^t$, does M produce, for all $a \in N$, a positive test for L_a in response to a finite sequence of length 1 consisting of the number 3^a.

Proof of the Lemma. Otherwise, the following procedure would exhibit a positive test for $\overline{K}$: to positively test $a \in \overline{K}$, present 3^a to M and put 2^a into the resulting positive test, p; by definition of L, p with input 2^a will eventually halt with output 1 if and only if $a \in \overline{K}$.

In view of the lemma, given any $M \in M_L^t$, there is a $b \in N$ such that for all texts, t, for L_b that begin with 3^b the convergence length for M on t exceeds 1. Thus, given any $M \in M_L^t$ we may construct an $M' \in M_L^t$ such that M' is at least as efficient as M but not the converse: M' simulates the behavior of M except for the input 3^b to which M' reacts immediately and correctly. Thus M is not maximally efficient. □

As a corollary to Proposition 3, there exist text-identifiable classes of languages for which no most efficient learning-machine exists.

Consideration of efficiency suggests other indices of the computational feasibility of text-identification. Natural measures include (a) the total number of computational steps prior to convergence, and (b) the maximum amount of internal memory-space filled at any time prior to convergence. For both of these performance measures results parallel to Proposition 3 are available.

3.4 Reliable Text-Identification

For a machine, M, to text-identify a class, L, of languages, it is sufficient that, for all $L \in L$, M converge to a positive test for L whenever a text for L is presented; M's behavior on texts for languages outside of *L* is not prescribed. In particular, for $L \notin \mathcal{L}$, M may well converge to a positive test for $L' \neq L$; that is, M may converge to an incorrect conjecture.

A learning machine, M, is called *reliable* just in case for every text, t, and for any language, L, either M converges on t to a positive test for L, or else M diverges on t.[11] Such discriminating behavior is "ideal" from the point of view of the information conveyed by a learning-machine. Reliable machines do not, misleadingly, continue to put out the same, incorrect hypothesis.

A machine, M, that responds to any finite sequence, s, by putting out a positive test for Rng(s) is an example of a reliable learning-machine: M text-identifies every finite language, and any text for an infinite language will cause M to change its conjectures indefinitely, hence, to diverge. Unfortunately, the only reliable machines are those that are restricted to text-identifying finite languages.

Proposition 4. If a machine, M, text-identifies any infinite language, then M is not reliable.

Proof. The result follows immediately from the lemma of Proposition 2. □

SECTION 4: INFORMANT-IDENTIFICATION

In this section we present a second paradigm of learning, to be called "informant-identification;" its definition is based on Gold (1967).

4.1 Languages and Tests

Learning machines are identified, as before, with ordinary Turing Machines; these machines examine languages and conjecture grammars. However, the languages and grammars are conceived more narrowly than before, and the means of presenting languages to learning machines assumes a different character.

Grammars are identified with *(positive and negative) tests.* A test is a procedure, P, such that for some language, L, P behaves as follows for all $x \in N$: given x as input, P eventually halts with output 1 if and only if $x \in L$, and P eventually halts with output 0 if and only if $x \notin L$. In this case, P is said to be *for* L. Notice that an arbitrary Turing Machine may fail to implement a test since it may, on certain inputs, halt with an output different from 0 to

[11]The notion of reliability was first studied by Minicozzi (see Blum & Blum, 1975).

1, or never halt at all.[12] Since Turing Machines provide a convenient normal form for both tests and positive tests, learning-machines may be conceived as announcing the programs of other machines; but we shall continue to refer to the latter as tests or positive tests.

Since the present learning paradigm concerns the identification of languages for which tests exist, we restrict attention (until Section 6) to such languages. Languages for which there are tests are called *recursive.*

4.2 Informants

In the new paradigm, the piecemeal presentation of a language to a learning-machine is called an *informant;* informants provide more information about a language than do texts. Let L be a language. An *informant for* L is any infinite sequence of ordered pairs of form $<x, 0>$ and $<x, 1>$ (repetitions allowed) such that for all $x \in N$, $<x, 1>$ appears in the sequence if and only if $x \notin L$, and $<x, 0>$ appears in the sequence if and only if $x \notin L$. An informant for a language, L, thus provides complete information about both membership and nonmembership in L (texts provide direct information about membership only).

A finite sequence that constitutes an initial segment of an informant, i, is said to be *in* i. We let *SEQ** stand for the set of all possible initial segments of informants.

4.3 Convergence and Informant-Identification

Each learning-machine, M, constitutes a possibly partial function from SEQ* into the set of positive tests. The behavior of a learning-machine in response to an informant may be conceived similarly to its behavior in response to a text. As before, if M fails to halt on a certain finite sequence, s, in a given informant, i (i.e., M puts out neither test nor positive test in response to s), then no more of i is put into M.

Given a learning-machine, M, an informant, i, and a test, t, we say that M *converges to* t *on* i just in case (a) M yields a test on every finite sequence in i, and (b) there is an n such that for all finite sequences, s, in i such that the length of s exceeds n, M applied to s yields t.[13] If M fails to converge to any test on a given informant, i, then M is said to *diverge* on i. M is said to *informant-identify* a language, L, just in case for every informant, i, for L there is a test, t, for L such that M converges to t on i.

[12] In contrast, every Turing Machine represents a positive test for some language (possibly the empty language). It is also worth noting that, as before, if there is one test for a given language, then there are an infinity of distinct tests for that language.

[13] Notice that according to our definition, a machine diverges on any informant that causes it to produce a (mere) positive test. This convention seems most analogous to that governing text-identification.

Let L be a class of languages. M is said to *informant-identify* L just in case M informant-identifies every language in L; in this case, L is called *informant-identifiable.* The set of machines that informant-identifies a given class, L, of languages is denoted: M^i_L.

SECTION 5: IDEAL INFORMANT-IDENTIFICATION

In this section the four senses of "ideal" considered in Section 3 are adapted to informant-identification. Theorems parallel to those of Section 3 are provided. All definitions will be relative to informant-identification.

5.1 Complete Informant-Identification

A learning-machine, M, that can informant-identify every (recursive) language would be ideal in an obvious sense. Such machines are called *complete (for informant-identification).* The following proposition shows that there are no ideal learning machines in this sense. Let R be the class of all recursive languages.

Proposition 5 (Gold, 1967). R is not informant-identifiable.

Proof. Let a learning-machine, M, be given. Define:

$$W = \{p : (\exists s \epsilon SEQ^*)\ (M(s) = p)\}$$

Since SEQ* is recursive, W is recursively enumerable. By a simple diagonal argument, no set of tests for all the recursive languages can be recursively enumerable. Therefore, if every member of W is a test, then M does not identify every recursive language. So, suppose that for some $p \epsilon W$, p is a (mere) positive test. Then, for some $s \epsilon SEQ^*$, $M(s) = p$. But then Rng(s) is a recursive language not informant identified by M since for some finite sequence in some informant for Rng(s), M fails to produce a test. Hence, M does not informant-identify R. □

Parallel to the case of text-identification, Proposition 5 shows that our conception of "ideal learning-machine (for informant-identification)" must be weakened. The next subsection considers a plausible way of attempting this.

5.2 Maximal Informant-Identification

A learning-machine, M, is said to be *maximal (for inforrmant-identification)* just in case there is some class, L, of languages such that M informant-identifies every superset of L that is informant-identifiable; in this case, M is said to be maximal *for* L. Thus, a maximal learning-machine informant-identifies a collection, L, of languages such that no proper superset of L is informant-identifiable; the learning ability of a maximal learning-machine is therefore unsurpassed by any other learning-machine. Such a device would be "ideal for L" in a strong sense.

Unfortunately, in the context of informant-identification, maximality is a vacuous notion.

Proposition 6. Let M be a learning-machine, and let L be a class of languages that M informant-identifies. Then, there is another machine, M′, that informant-identifies a collection, L', of languages such that $L \subsetneq L'$, and M does not informant-identify L'.

Proof. Let t be a test for a recursive language not in L (by Proposition 5 there must be such a test). On every informant, i, M′ stores input pairs in memory, and conjectures t until t becomes inconsistent with an initial segment of i put into M′. In the latter case M′ simulates M on i, using stored pairs in order to present all of i to M. □

So, there are no maximal learning-machines for informant-identification.

5.3 Efficient Informant-Identification

In the context of informant-identification, the notions of "convergence length," "most efficiency," and "maximal efficiency" carry over straightforwardly from Section 3.3. And, similarly to before, whereas there are most efficient machines for the informant-identification of certain classes of languages, there are also informant-identifiable classes of languages for which there are not even maximally efficient learning machines.

Proposition 7. There are informant-identifiable collections of languages that cannot be informant-identified by any maximally efficient learning machine.

Proof. The class, L, in the proof of Proposition 3 is one such collection of languages. The argument is parallel to that given before. □

As a corollary to Proposition 7, there exist informant-identifiable classes of languages for which no most efficient learning-machine exists.

Results parallel to Proposition 7 obtain for other natural indices of the computational feasibility of informant-identification.

5.4 Reliable Informant-Identification

A learning-machine is called *reliable (for informant-identification)* just in case for every informant, t, for any recursive language, L, either M converges on t to a test for L, or else M diverges on t. Reliable machines are available for every informant-identifiable class of languages.

Proposition 8. If a class, *L,* of languages is informant-identifiable, then there is a reliable learning-machine that informant-identifies *L*.

Proof. Let $M \epsilon \mathcal{M}^{i}_{L}$. We construct a reliable machine, $M' \epsilon \mathcal{M}^{i}_{L}$. M′ internally simulates M on every informant, i. On each finite sequence, s, in i, M′ puts out the conjecture, c, made by M on s, but only after M′ has verified that c is consistent with s. If M puts out no conjecture on s, or if c is not defined on all the numbers appearing in left-coordinates of s, then both M and M′ diverge on i. If c is found to be inconsistent with s, then M′ puts out a test for {n}, where n is the length of s. M′ is easily seen to identify *L,* and to be reliable. □

SECTION 6: CONCLUSION

We summarize the preceding discussion in Table 1. Only for reliable informant-identification are ideal machines generally available. For the seven other senses of "ideal" that we have considered, such learning-machines are either not generally available, rarely available, or nonexistent altogether. These facts suggest that the search for ideal learning-machines, in any natural sense of "ideal" is not a fruitful strategy for Artificial Intelligence or Cognitive Science.

TABLE I
Summary of Results Concerning Ideal Learning-Machines

	Learning Paradigm	
Sense of ideal	Text-identification	Informant-identification
Complete	Nonexistent	Nonexistent
Maximal	Exist only for collections of finite languages	Nonexistent
Efficient	Exist for some but not all text-identifiable collections of languages	Exist for some but not all informant-identifiable collections of languages
Reliable	Exist only for collections of finite languages	Exist for every informant-identifiable collection of languages

On the other hand, there is a related endeavor of potential empirical interest. Instead of searching for machines of unsurpassed learning ability, one might instead investigate the limitations inherent in different kinds of learning devices. To clarify, let us call any subset of Turing Machines a *(learning) strategy*. Some strategies (i.e., subsets of machines) have no natural characterization, but others correspond to plausible properties of the human system for acquiring natural language. For example, one strategy includes just those machines that seem unable to "remember" linguistic input presented long ago, and another strategy includes the machines that never abandon a conjecture that is consistent with all the data seen to that point. A strategy, S, is called *restrictive (for text-identification)* just in case there is a text-identifiable collection, L, of languages such that no machine in S can text-identify L. Thus, if children can be shown to embody a learning procedure that belongs to a restrictive strategy, the class of natural languages would be limited thereby. In this way, investigation of restrictive strategies might facilitate the development of an explanatory theory of natural language. These issues are examined in detail in Osherson, Stob, and Weinstein (1981).

Finally, we note that one kind of less-than-ideal learning machine lowers its standards for accuracy, and is content to approximate an input language. Within this paradigm, identification does not require convergence to a grammar for precisely the input language; a "near miss" suffices. Several formalizations of approximate learning are available. Osherson and Weinstein (1982) study the effects of weakening the definition of identifiability in these ways.

REFERENCES

Blum, L., & Blum, M. Toward a mathematical theory of inductive inference. *Information and control*, 1975, *28*, 125–155.

Boolos, G., & Jeffrey, R. *Computability and logic* (2nd ed.). Cambridge: Cambridge University Press, 1980.

Gold, E. M. Language identification in the limit. *Information and control*, 1967, *10*, 447–474.

Osherson, D., Stob, M., & Weinstein, S. *Learning strategies*. Manuscript, 1981.

Osherson, D., & Weinstein, S. *Criteria of learning*. Manuscript, 1982.

Rogers, H. *Theory of recursive functions and effective computability*. New York: McGraw-Hill, 1967.

Wexler, K., & Culicover, P. *Formal principles of language acquisition*. Cambridge, MA: MIT Press, 1980.

Winston, P. H. Learning by creating and justifying transfer frames. *Artificial Intelligence*, 1978, *10*, 147–172.