Machine Learning, 19, 133–152 (1995)
© 1995 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Comprehension Grammars Generated from Machine Learning of Natural Languages*

PATRICK SUPPES suppes@csli.stanford.edu
CSLI, Ventura Hall, Stanford University, Stanford CA 94305-4115

MICHAEL BÖTTNER boettner@ockham.stanford.edu
Max-Planck-Institut für Psycholinguistik, 6526 XD Nijmegen, The Netherlands

LIN LIANG liang@csli.stanford.edu
CSLI, Ventura Hall, Stanford University, Stanford CA 94305-4115

Editor: Jaime Carbonell

Abstract. We are developing a theory of probabilistic language learning in the context of robotic instruction in elementary assembly actions. We describe the process of machine learning in terms of the various events that happen on a given trial, including the crucial association of words with internal representations of their meaning. Of central importance in learning is the generalization from utterances to grammatical forms. Our system derives a comprehension grammar for a superset of a natural language from pairs of verbal stimuli like *Go to the screw!* and corresponding internal representations of coerced actions. For the derivation of a grammar no knowledge of the language to be learned is assumed but only knowledge of an internal language.

We present grammars for English, Chinese, and German generated from a finite sample of about 500 commands that are roughly equivalent across the three languages. All of the three grammars, which are context-free in form, accept an infinite set of commands in the given language.

Keywords: learning, natural language, grammatical generalization, robotics

1. Introduction and Overview

We are developing a theory of machine language learning in the context of robotic instruction of elementary assembly actions. More specifically, our situations of language learning concern actions like moving to objects, picking up objects and moving objects in environments. Typical objects are screws, nuts, washers, and sleeves of various colours, sizes, and shapes, related to each other by common spatial relations. Typical commands to be comprehended are *Get a screw!, Go to the black washer behind the screw!* and *Put the screw in the left hole!* The system developed so far deals successfully with English, Chinese, and German, as well as several other languages, which for a lack of space we will report on separately.

The overlap between the utterances that a person is able to understand and those that a person is able to produce may be small. A comprehension grammar would, e.g., be able to derive *Go to the left of the nut screw!* but not *Put a nut the near screw!* although neither of these utterances should be derivable in an English production grammar. A purely comprehension grammar should parse a superset of the set of strings we would expect a production grammar to derive.

*The results reported were originally presented at the Eighth Amsterdam Colloquium, December 17–20, 1991.

Our axioms, given in Section 2, for machine learning of comprehension are long and complicated, even though we analyze here the learning of relatively small fragments of the natural languages considered, but they do faithfully reflect the computer implementation, as well as extend the axioms given in our earlier paper (Suppes, Liang, & Böttner, 1992). We do believe the learning principles expressed in the axioms can serve as the basis for learning much more extended fragments and represent concepts that must be present in any system that starts with no knowledge of the natural language to be learned. The axioms naturally fall into three parts. First are those concerning computations using short-term memory based on learned associations; second, those dealing with changes in the state of long-term memory, which include computations of grammatical generalizations updating the comprehension grammar and computations of denoting value; and third, the axiom describing the process of comprehension when more learning is not required to understand a particular command.

In Section 3 we describe the internal language, which is not learned, and then in Section 4 we provide an extensive commentary on the axioms, using, as would be expected, the resources of the internal language. In Section 5 we present detailed results on the comprehension grammars generated, in Section 6 we state a number of significant problems we have not yet solved, and in Section 7 we review related work.

The results for the comprehension grammars of English, Chinese and German are linguistically surprising. Once congruence computations (see Axiom 2.5) are made, 9 grammatical rules are common to all three languages, 2 to Chinese and English, 7 to English and German, and 0 for Chinese and German. The surprise is that no special rules for English are required. Seven special rules are required for Chinese and 4 for German. As these numbers suggest and the details confirm, our comprehension grammars are vastly simpler than the corresponding production grammars.

2. Learning System

What we want to describe now is the process of learning in terms of the various events that happen on a given trial. Initially the robot has no knowledge of the lexicon or grammatical structure of the language to be learned. The robot begins a trial in a given state of memory. There are four parts of this memory that are changed due to learning: the probabilistic association relation, the denotational values of words, the set of grammatical forms and short-term memory of the current command and association computations.

1. The first is the association relation between words of a given language and internal symbols that represent denotations of members of categories in the internal representation language. The semantic categories of action, object, property and spatial relation, as well as the internal language (Lisp), are given in advance and not learned. For example, the action of putting will be represented internally, let us say, by the symbol $\$p$ and will have three different linguistic representations in English, German and Chinese. The problem will be to learn in each language what word is properly associated with the internal representation $\$p$. The same process of association must take place for the internal symbols for objects, properties and spatial relations (Axiom 1.2). But knowing such associations is not enough, contrary to some naive theories of associationism.
2. It is also important to have grammatical forms in long-term memory (Axiom 2.4). We will not try to lay out everything about grammatical forms that we have in our fully stated

theory, but it will be useful to give some examples. Consider the verbal command *Get the screw!* This would be an instance of the grammatical form A_1 *the OBJ!*, where A_1 is a category of actions and *OBJ* is the category of objects. As might be expected we do not actually have just a single category of actions, but several subcategories, depending upon the number of arguments required, etc. The central points here, however, are that: (i) the grammatical forms are derived only by generalization from actual instances of verbal commands given to the robot, and (ii) the grammatical forms are used in parsing any new command.

3. The third part of memory that varies is the short-term memory that holds a given verbal command for the period of the trial on which it is effective. This memory content decays and is not available for access after the trial on which a particular command is given (Axiom 1.7). This is also true of any association computations (Axioms 1.1–1.6).

When a new verbal command is given to the robot, the program tries to understand the command by parsing it and constructing an internal representation (Axiom 3). For example, if mistakenly the word *get* had been associated to $\$s$, the internal symbol for *screw*, and *screw* had been associated with $\$g$, the internal symbol for *get*, then the grammatical form that would have been generated and now found upon a second occurrence of the verbal command would be *OBJ the* A_1*!* Now if there were no such grammatical form present in long-term memory, once the associations were formed such a grammatical form would be created by the process of generalization (Axiom 1.3), and a corresponding grammatical rule for the natural language generated (Axiom 1.4).

The important case of learning is when there is no parse of the given command, i.e., no grammatical form can be generated recursively from the grammatical forms in memory to match the form of the given verbal command. In this case the robot is unable to make a response. The correct response must be coerced (Axiom 1.1). On the basis of this coercion, a new internal representation is formed. In computer simulations, coersion consists of providing the program the correct internal representation of the presented command. At this point the critical step comes from probabilistic association between words of the verbal utterance and the internal denoting symbols of the new internal representation (Axiom 1.2). From the association between utterance and internal representation, one or more new grammatical forms are generated (Axiom 1.3), as well as one or more new grammatical rules (Axiom 1.4). Possibly because of the new associations at least one of the old grammatical forms and rules is deleted, for, according to our axioms, a word or internal denoting symbol can have exactly one association, so when a new association is formed for a word any old associations must be deleted (Axioms 2.7 and 2.8). Also computed are the new denotational values of words occurring on the trial (Axiom 2.3).

The learning process as described so far must be extended to the dynamic computations, recursive in nature, for updating the grammar as new commands are encountered that cannot be parsed by the existing grammar. Axiom 1.6 formulates the recursive procedure for generating new associations of parts of a grammatical form with parts of the associated internal form. Axioms 2.4 and 2.8 then characterize the computations for dynamically changing the current grammar by adding a new rule or deleting a now redundant old rule.

2.1. Learning Principles

To facilitate the detailed statement of principles governing the learning process just described, certain notational conventions are useful. First, generally we use Latin letters to

refer to verbal commands or their parts, whatever the natural language, and we use Greek letters to refer to internal representations or their parts. The letters a, a_i, a_i', etc. refer to words in a verbal command, and the Greek letters α, α_j, α_j', etc. refer to internal denotations. The Roman letter t, as well as t_i, t_i' refer to terms of a verbal command, and correspondingly, τ, τ_i, τ_i' to terms of an internal representation, i.e., in the present setup, Lisp expressions. The symbols s, s', and $s(t)$, showing that t is a term of s, refer to entire verbal commands, and correspondingly σ, σ', $\sigma(\tau)$ to entire internal representations. Grammatical forms—either sentential or term forms—are denoted by g or also $g(X)$ to show a category argument of a form; correspondingly the internal representations of a grammatical form are denoted by γ or $\gamma(X)$. We violate our Greek-Latin letter convention in the case of semantic categories or category variables X, X', Y, etc. We use the same category symbols in both grammatical forms and their internal representations. In order to avoid enumeration of cases, when we use $\gamma(X)$ or $\tau(X, Y)$, for example, it is not required that X, or X and Y, occur as free variables in the grammatical form and associated internal representation.

Axioms of Learning

1. Association Computations using Working Memory

   1.1 *Association by contiguity.* If on any trial verbal command s is contiguous with a coerced action that has internal representation σ, then s is associated with σ, i.e., in symbols $s \sim \sigma$.

   1.2 *Probabilistic association.* On any trial n, let s be associated to σ in accordance with Axiom 1.1, let $\{a_i\}$ be the set of words of s not associated to any internal denoting symbol of σ, let $d_n(a_i)$ be the current denotational value of each such a_i and let $\{\alpha_j\}$ be the set of internal denoting symbols not currently associated with any word of s. Then

   i an element α_j is uniformly sampled without replacement from $\{\alpha_j\}$,

   ii at the same time an element a_i is sampled without replacement from $\{a_i\}$ with the sampling probability

$$p_n(a_i) = \frac{d_n(a_i)}{\sum_{\{a_i\}} d_n(a_i)},$$

   iii the sampled pairs are associated, i.e. $a_i \sim \alpha_j$,

   iv sampling continues until either the set $\{a_i\}$ or the set $\{\alpha_j\}$ is empty.

   1.3 *Form generalization.* If at any step of an association computation on any trial, $g(t(Y)) \sim \gamma(\tau(Y))$, $t(Y) \sim \tau(Y)$ and $\tau(Y)$ is derivable from X, then $g(X) \sim \gamma(X)$.

   1.4 *Grammar-rule generation.* If at any step of an association computation on any trial, $t(Y) \sim \tau(Y)$ and $\tau(Y)$ is derivable from X, we infer for the natural language the grammatical rule $X \rightarrow t(Y)$.

   1.5 *Form specification.* If at any step of an association computation on any trial, $g(X) \sim \gamma(X)$, $t(Y) \sim \tau(Y)$ and $\tau(Y)$ is derivable from X, then $g(t(Y)) \sim \gamma(\tau(Y))$.

   1.6 *Form association.* Let $g \sim \gamma$ at any step of an association computation on any trial.

   (a) If X occurs in g and $(fo_1\ X\ *)$ occurs in γ, then $X \sim (fo_1\ X\ *)$.

(b) If

i wX is a substring of g with $g \sim \gamma$ such that $w = a$, which is a nondenoting word, or if X is preceded by a variable, or is the first symbol of g, $w = \varepsilon$, the empty symbol, and

ii $\tau(X)$ is the maximal Lisp form of γ, containing the occurrence of X and no other occurrence of denoting categories,

then $wX \sim \tau(X)$.

(c) If

i $X_1 w_1 \cdots w_{m-1} X_m$ is a substring of g, where the X_i, $i = 1, \ldots, m$ are not necessarily distinct category names and w_i are substrings, possibly empty, or words that have no association to internal symbols on the given trial,

ii $\tau(X_{\pi(1)}, \ldots, X_{\pi(m)})$ is the minimal Lisp form of γ containing $X_{\pi(1)}, \ldots, X_{\pi(m)}$,

then $X_1 w_1 \cdots w_{m-1} X_m \sim \tau(X_{\pi(1)}, \ldots, X_{\pi(m)})$, where π is a permutation of the numbers $1, \ldots, m$.

1.7 *Delete contents.* The content of working memory is deleted at the end of each trial.

2. Changes in State of Long-term Memory

2.1 At the beginning of trial 1, the association relation $\sim$, the congruence relation $\approx$ and the set of grammatical rules G are empty. Moreover, the denotational value $d(a_i)$ is the same for all words a_i.

2.2 The semantic categories and the semantically interpreted internal language are stored in long-term memory and remain unchanged from trial to trial.

2.3 *Denotational learning computations.* If at the end of trial n a word a_i in the presented verbal stimulus is associated with some denoting internal symbol α_j of the internal representation σ of s at the end of the trial, then

$$d_{n+1}(a_i) = (1 - \theta)d_n(a_i) + \theta,$$

and if a_i is not so associated

$$d_{n+1}(a_i) = (1 - \theta)d_n(a_i).$$

Moreover, if a word a_i does not occur on trial n, then

$$d_{n+1}(a_i) = d_n(a_i),$$

unless the association of a_i to an internal symbol α_j is broken on trial n, in which case

$$d_{n+1}(a_i) = (1 - \theta)d_n(a_i).$$

2.4 *Grammar computations.*

(a) If on trial n, $g \sim \gamma$ on the basis of Axiom 1.6, then

i if no substring g' of g is already in long-term memory with $g' \sim \gamma'$, then $g \sim \gamma$ is stored in long-term memory,

ii if a substring g' of g is already in long-term memory with $g' \sim \gamma'$, then g and γ are reduced to $g(X) \sim \gamma(X)$ where X is the category generating γ',

iii this recursive reduction continues until no further reduction is possible,

iv the reduced g, call it rg, together with $r\gamma$, i.e., $rg \sim r\gamma$, is added to long-term memory if not already present, as is the corresponding grammatical rule generated by Axiom 1.4.

(b) If $g \sim \gamma$ is in long-term memory at the beginning of trial n but a substring $g' \sim \gamma'$ of g is generated on the basis of Axiom 1.6, then $rg(X) \sim r\gamma(X)$ is stored in memory and g, γ and $g \sim \gamma$ are deleted from long-term memory, as is the grammatical rule.

2.5 *Congruence computations.* If w is a substring of g and w' a substring of g' and are such that

(a) $g \sim \gamma$ and $g' \sim \gamma$,

(b) g and g' have only category symbols as denoting terms,

(c) g' differs from g only in the occurrence of w' in place of w,

(d) w and w' contain no category symbols,

then $w' \approx w$ and the congruence is stored in long-term memory.

2.6 *Form memory trace.* The first time a form generalization (Axiom 1.3), grammatical rule (Axiom 1.4) or congruence (Axiom 2.5) is formed, the word associations on which the generalization, grammatical rule or congruence is based are stored with it in long-term memory.

2.7 *Delete prior associations.* When a word in a verbal command or an internal symbol is given a new association, any prior associations of that word or symbol are deleted from long-term memory.

2.8 *Delete form association or grammatical rule.* If $a \sim \alpha$ is deleted, then any form generalization, grammatical rule or congruence for which $a \sim \alpha$ is a memory trace is also deleted from long-term memory.

3. Comprehension-and-Response axiom

If command s occurs at the beginning of trial n, then the following:

(a) Using a standard context-free parser Π on the set G_n of grammatical rules in long-term memory, a parse of s is attempted.

(b) If exactly one parse of s is found, the parse tree of s is used to construct the internal representation σ of s by replacing each grammatical rule by the corresponding derivation (Axiom 1.4) of the internal language, and σ is then executed.

(c) If more than one parse is found, say, the set $\Pi(s)$ of parses, then each associated internal representation σ is tested for compatibility with the given physical environment E_n, and parses incompatible with E_n are deleted from the set $\Pi(s)$; one parse is selected at random from the reduced set $\Pi'(s)$ and the associated σ is executed if possible; if execution is not possible, no other parses are sampled.

(d) If no parse is found, no response is made.

We comment in detail on the learning process described by the axioms after the next section on the internal language.

3. Internal Language

The internal language is a Lisp regimentation of English lexical semantics, but not, of course, English syntax. For example, the internal symbols for spatial relations correspond more closely to the semantics of English prepositions than to that of any of the other languages we have considered. It is an objective of our work to eliminate in the future this English lexical bias. We believe that much can be accomplished in this direction by forming for each set of concepts a partition fine enough to cover most if not all the languages we would be likely to consider. This is not to suggest that such partitions solve a host of other problems. Internal representations will therefore be Lisp-expressions. A Lisp-expression is any string $(FE_1, \ldots, E_n)$ where F is a function symbol and $E_1, \ldots, E_n$ are either atoms or Lisp-expressions. (For readability we deviate from standard Lisp printing conventions by using italics and subscripts.) The atoms refer to properties, actions, spatial relations, and objects of the environment and fall into corresponding categories. In particular, we have category P of properties like $\$small$, $\$black$, $\$square$; category OBJ of special properties serving to single out objects like $\$screw$, $\$nut$, $\$washer$, $\$plate$, $\$hole$, $\$sleeve$; category R of binary relations like $\$front$, $\$back$, $\$left$, $\$right$; category S of the set of objects in the environment, which is denoted by $*$; action categories: A_1 like $\$get$, $\$open$, $\$close$, A_2 like $\$go$, A_3 like $\$put$.

In addition to these we have categories of expressions that are not atoms: category A of actions like $(fa_1\ \$get\ (io\ (fo_1\ \$screw)))$; category S of objects like $(fo_1\ \$screw\ *)$; category G of regions like $(fr_1\ \$near\ (io\ (fo_1\ \$screw\ *)$.

Expressions denoting subsets of these categories are derived by the semantic operations listed below, but commitment to a set-theoretical framework is not essential. The lexical rules of the internal language are omitted in this paper.

Grammar of Internal Language

1.	fo_1 (focus-on-object-1)	$S \rightarrow (fo_1\ OBJ\ *)$
2.		$S \rightarrow (fo_1\ P\ S)$
3.		$S \rightarrow (fo_1\ G\ S)$
4.	fo_2 (focus-on-object-2)	$S \rightarrow (fo_2\ R\ S)$
5.	*io* (identify object)	$O \rightarrow (io\ S)$
6.	*so* (select-object)	$O \rightarrow (so\ S)$
7.	fa_1 (form-action-1)	$A \rightarrow (fa_1\ A_1\ O)$
8.	fa_2 (form-action-2)	$A \rightarrow (fa_2\ A_2\ G)$
9.	fa_3 (form-action-3)	$A \rightarrow (fa_3\ A_3\ O\ G)$
10.	fa_4 (form-action-4)	$A \rightarrow (fa_4\ A_4\ O)$
11.		$A \rightarrow (fa_4\ A_4\ D\ O)$
12.	*fdir* (form-direction)	$D \rightarrow (fdir\ R)$
13.	fr_1 (form-region-1)	$G \rightarrow (fr_1\ R\ O)$
14.	*and* (form-and-property)	$P \rightarrow (and\ P\ P)$
15.	*or* (form-or-property)	$P \rightarrow (or\ P\ P)$
16.	*not* (form-not-property)	$P \rightarrow (not\ P)$

The robot's internal language is the set of commands generated by the grammar. Notice that the operations also return expressions of our initial categories, in particular P and S which means that these categories are recursive. So we will have in our internal language

symbols for each content word of a command. The English command *Get a screw!* has the following internal representation $(fa_1\ \$g\ (so(fo_1\ \$s\ *)))$, where $\$g$ and $\$s$ occur in the internal language as counterparts of the English content words *get* and *screw*. The internal representation given above has three more symbols: fo_1, *so*, and fa_1. These symbols are abbreviations of the semantic operations *focus-on-object*, *select-object*, and *form-action*. The purpose of these operations is to provide what we think is the procedural structure of the natural language commands mentioned above. Before it can perform the action denoted by $\$g$, the robot has to determine the object of this action. The object that is intended to become the object of the action is returned by the operation *so*. This operation is nondeterministic: it just selects one out of a set of objects. The input for *so* is a set of objects. This set of objects is the output of the operation fo_1: it takes a property and the set of objects in the robot's environment and returns a set of objects. The set of objects in the robot's environment is represented by the symbol $*$ in our internal language. In our example, the property the environment is checked for is the presence of screws. The procedure *so* then takes this set of all screws of the environment as input and returns an arbitrary screw from this set. This particular screw together with the action $\$g$ is the input of the operation fa_1.

To illustrate our internal language we give some examples for basic types of action and object structures together with their associated English counterparts.

Action Structures

$$\begin{aligned}
\textit{Get the screw!} &\sim (fa_1\ \$get\ (io\ (fo_1\ \$screw\ *))) \\
\textit{Go near the screw!} &\sim (fa_2\ \$go\ (fr_1\ \$near\ (io\ (fo_1\ \$screw\ *)))) \\
\textit{Put the screw behind the nut} &\sim (fa_3\ \$put\ (io\ (fo_1\ \$screw\ *)) \\
&\qquad (fr_1\ \$behind\ (io\ (fo_1\ \$nut\ *))))
\end{aligned}$$

Object Structures

$$\begin{aligned}
\textit{screw} &\sim (fo_1\ \$screw\ *) \\
\textit{large screw} &\sim (fo_1\ \$large\ (fo_1\ \$screw\ *)) \\
\textit{screw which is large} &\sim (fo_1\ \$large\ (fo_1\ \$screw\ *)) \\
\textit{screw which is not large} &\sim (fo_1\ (not\ \$large)\ (fo_1\ \$screw\ *)) \\
\textit{large black screw} &\sim (fo_1\ \$large\ (fo_1\ \$black\ (fo_1\ \$screw\ *))) \\
\textit{large and black screw} &\sim (fo_1\ (\text{and}\ \$large\ \$black)\ (fo_1\ \$screw\ *)) \\
\textit{left screw} &\sim (fo_2\ \$left\ (fo_1\ \$screw\ *)) \\
\textit{left square screw} &\sim (fo_2\ \$left\ (fo_1\ \$square\ (fo_1\ \$screw\ *))) \\
\textit{screw behind a nut} &\sim (fo_1\ (fr_1\ \$behind\ (so\ \ (fo_1\ \$nut\ *)))\ (fo_1\ \$screw\ *)) \\
\textit{large screw behind a nut} &\sim (fo_1\ \$large\ (fo_1\ (fr_1\ \$behind\ (so\ \ (fo_1\ \$nut\ *))) \\
&\qquad (fo_1\ \$screw\ *)))
\end{aligned}$$

Putting object structures into action structures we may derive structures of arbitrary degrees of complexity, as e.g.,

$$(fa_3\ \$put\ (io\ (fo_1\ (fr_1\ \$near\ (io\ (fo_1\ \$nut\ *)))\ (fo_1\ \$screw\ *)))$$

$$(fr_1\ \$behind\ (io\ (fo_2\ \$left\ \ (fo_1\ \$hole\ *)))))$$

which corresponds to the English command

Put the screw that is near the nut behind the left hole!

The operations fo_1 and fo_2 have in common that they focus the robot's attention on objects of a certain kind in the set of objects that can be perceived by the robot on the occasion of being given a particular command: fo_1 takes a property and a set of objects and returns the objects that are referred to by the element of *OBJ*. So the expression

$$(fo_1\ \$screw\ *)$$

returns the set of all objects from the set in the robot's perceptual environment that are screws. In

$$(fo_1\ \$black\ (fo_1\ \$screw\ *))$$

fo_1 operates recursively: it first returns the set of all screws in the robot's environment and then returns the subset of all black screws. The operation fo_2 focuses on a set of objects that bear another certain relation to a set of objects. A typical example is the English expression *left screw* which, in our internal language, is given the structure

$$(fo_2\ \$left\ (fo_1\ \$screw\ *)).$$

A region is a set of locations that stand in a certain relation to at least one object. Regions are built from a binary spatial relation and an object by fr_1. The operation *so* selects an arbitrary object from a set of perceptually given objects. The operation *io* identifies a unique object satisfying certain properties in a set of perceptually given objects. Not surprisingly a number of operations arise in the context of actions. This has to do with the various subclasses of actions depending on the different valencies exhibited by actions.

4. Detailed Comments on the Axioms

An important and distinguishing feature of the comprehension grammars we generate is that the denoting words of a language, as defined just above, are assigned to semantic rather than syntactic categories. For example, the English word *left* is in the category of spatial relations independent of its differing syntactic roles in the following phrases: *Move left! Go to the left! Pick up the left screw! Pick up the screw which is left of the box!* Similar examples appear in the other languages. Moreover, the nondenoting words are not assigned to any category at all except in the formal sense of the general category of being nondenoting, so that the usual syntactic distinctions that would apply to such words are not made. On the other hand, nondenoting words occur in different roles in grammatical forms.

4.1. Creativity

To get a better understanding of the axioms let us consider two examples that show that the grammars generated can comprehend new commands and even an infinite number of new commands. The first is to show that sentences can be understood that are new to the learner. The second is to show that the grammar can analyse sentences that are longer than any of the sentences of the finite sample of the language to be learned.

Predication Test. Once the generalized associations are established the internal representations of new commands can be derived. For instance, on the basis of having been learned correctly, *Get the screw!*, *Go to the nut!*, and *Get the nut!*, the following command can be understood by the comprehension grammars, and thus executed by the robot: *Go to the screw!*

Recursivity. Our system does not just derive new structures of the same length but is also able to cope with recursivity by understanding commands longer than any of the commands occurring in the learning sequence. On the basis of having learned correctly

Get the nut! (1)

Get the plate! (2)

Get the red washer! (3)

Get the nut which is left of the screw! (4)

Get the screw behind the nut! (5)

it would be able to comprehend the following command:

Get the red screw behind the plate which is left of the washer! (6)

Assume that, on the basis of previous experience, the robot's long-term memory holds the following associations:

$$\textit{nut} \sim \$n,\ \textit{get} \sim \$g,\ \textit{screw} \sim \$s,\ \textit{plate} \sim \$p,\ \textit{washer} \sim \$w,\ \textit{red} \sim \$r,\ \textit{behind} \sim \$b,\ \textit{left} \sim \$l. \quad (7)$$

Coerce:

$$\textit{Get the nut!} \sim (fa_1\ \$g\ (io\ (fo_1\ \$n\ *))) \quad (8)$$

$$\textit{Get the red washer!} \sim (fa_1\ \$g\ (io\ (fo_1\ \$r\ (fo_1\ \$w\ *)))) \quad (9)$$

$$\textit{Get the nut which is left of the screw!} \sim (fa_1\ \$g\ (io\ (fo_1\ (fr_1\ \$l\ (io\ (fo_1\ \$s\ *)))\ (fo_1\ \$n\ *)))) \quad (10)$$

$$\textit{Get the screw behind the nut!} \sim (fa_1\ \$g\ (io\ (fo_1\ (fr_1\ \$b\ (io\ (fo_1\ \$n\ *)))\ (fo_1\ \$s\ *)))) \quad (11)$$

Form Generalization applied to (8) extends memory by

$$A_1\ \textit{the}\ \textit{OBJ!} \sim (fa_1\ A_1\ (io\ (fo_1\ \textit{OBJ}\ *))) \quad (12)$$

Then by *Form Association* (Axiom 1.6a)

$$\textit{OBJ} \sim (fo_1\ \textit{OBJ}\ *) \quad (13)$$

And then by *Rule Generation* (Axiom 1.4) and IL Rule 1, i.e., Internal Language Rule 1, we get as a grammatical rule of the fragment of English

$$S \rightarrow \textit{OBJ} \quad (14)$$

Going back to (12) again by *Form Generalization* and (14) we obtain

$$A_1 \; the \; S \sim (fa_1 \; A_1 \; (io \; S)) \tag{15}$$

By *Form Association* again applied to (15) we get

$$the \; S \sim \; (io \; S) \tag{16}$$

and now from (15) by *Rule Generation* and IL Rule 5 we get the grammatical rule

$$O \rightarrow \; the \; S. \tag{17}$$

Using once again *Form Generalization* on (15) together with IL Rule 5, we have

$$A_1 \; O \sim (fa_1 \; A_1 \; O), \tag{18}$$

which by use of IL Rule 7 and *Rule Generation* produces the grammatical rule

$$A \rightarrow A_1 \; O \tag{19}$$

By very similar use of (9)–(11), we infer the following additional grammatical rules

$$S \rightarrow S \; G \tag{20}$$
$$S \rightarrow P \; S \tag{21}$$
$$S \rightarrow S \; which \; is \; G \tag{22}$$
$$G \rightarrow R \; O \tag{23}$$
$$G \rightarrow R \; of \; O \tag{24}$$

Using the grammatical rules (14), (17), (19)–(24) derived from (8)–(11), and the lexical rules derived from the lexical associations (7) and the categories of the internal symbols, we may parse and thereby comprehend the long test sentence (6) by using Axiom 3b.

4.2. *Contiguity*

Our learning algorithm is based on the idea that the structure of the internal language can be used to derive the constituent structure of natural language utterances. This idea easily invites the objection that the algorithm is successful for languages with only contiguous constituents but inevitably fails for languages with noncontiguous constituents. A case in point is the German command

Heb die Schraube hoch, die groß ist! (25)

Note that in this command *Schraube die groß ist* forms one constituent from the standpoint of our internal language. This constituent, however, is not contiguous since it is interrupted by the constituent *hoch*. Now there is in German also a way to express the command expressed by (25) with contiguous constituents:

Heb die Schraube, die groß ist, hoch! (26)

(We disregard for the moment that one could argue that (26) has also noncontiguous constituents since *heben* and *hoch* are forms of one word *hochheben*.) But (26) is considered

less natural and the algorithm should be able to come up with a grammar on the basis of both (25) and (26). In fact, our algorithm is able to derive a grammar in both cases. The grammar derived from (26) will have among its rules the following one:

$$A_4\ O\ ist\ D \sim\ (fa_4\ A_4\ D\ O) \tag{27}$$

The grammar derived from (25), however, will not have this rule but the following rule instead:

$$A_4\ die\ S\ D\ die\ P\ ist \sim\ (fa_4\ A_4\ (io\ (fo_1\ P\ S))\ D)$$

Note that (26) is less general than (25). All the rules specific to German are of that kind.

4.3. *Context-free, but not Regular Languages*

Our axioms can generate grammars for languages that are context-free but not regular. This may happen because of the presence of so-called center-embedding structures exhibited by the following German example:

1. *Nimm die nahe der Mutter befindliche Schraube!*
2. *Nimm die nahe der hinter dem Loch befindlichen Mutter befindliche Schraube!*
3. *Nimm die nahe der hinter dem vor der Platte befindlichen Loch befindlichen Mutter befindliche Schraube!*

...

Our axioms are able to deal with center-embedding structures by having the following associations in memory:

$$R\ der\ S'\ befindliche\ S \sim (fo_1\ (fr\ R\ (io\ S'))\ S).$$

This amounts to having the following rule in a grammar for German:

$$S \rightarrow R\ der\ S'\ befindliche\ S.$$

4.4. *Concept of Denotational Value*

In the probabilistic theory of machine learning of natural language which we have been developing, we have encountered in a new form a standard problem in the analysis of the semantics of natural language, namely, how to handle words that are nondenoting. We do not mean nondenoting in some absolute sense, but relative to our standard set of semantical categories. It may well be that in some elaborate set-theoretical semantics of natural language, nondenoting words like the definite article *the* denote a complicated set-theoretical function. We have something simpler and closer to the common man's view of what denotations are. We take as denoting, color words, common nouns, familiar concrete action words, etc.

When a child learning a first language or an older person learning a second language first encounters utterances in that new language, there is no uniform way in which nondenoting words are marked. There is some evidence that various prosodic features are used in English and other languages to help the child. For example, in many utterances addressed to very young children the definite or indefinite article is not stressed but rather the common noun it modifies, as in the expression *Hand me the cup!* But such devices do not seem uniform

Table 1. Selected denotational values.

English		Chinese		German	
word	d-value	word	d-value	word	d-value
plate	0.997	ban	0.999	Platte	0.998
nut	1.000	luomu	0.999	Mutter	0.999
large	0.967	dade	1.000	gross	1.000
place	0.986	fang	0.970	bring	0.989
in	0.972	jing	0.999	in	0.889
the	0.003	nage	0.007	die	0.002
a	0.003	yige	0.005	eine	0.002
		ba	0.001		
		na4	0.127		
		na4li	0.246		

and in any case are not naturally available to us in our machine-learning research, where we use written input of words without additional prosodic notation. A central feature of our approach to machine learning is the probabilistic association between words of the natural language being learned and internal symbols that denote, as explained earlier.

It is appropriate that at the beginning all words are treated equally and so the associations are formed from sampling based on a uniform distribution. On the other hand, after many words have been learned and a good piece of language has been acquired by the robot, it is very unnatural, and also inefficient, if the robot is now given the esoteric command *Get the astrolabe!* to have the internal symbol *$ast* to be associated with equal probability with the definite article *the* and *astrolabe*—we assume here that the association of *get* is already correctly fixed. After much experience, there should be very little chance of associating the definite article *the* with any denoting symbol.

To incorporate such learning, many variant learning models are easily formulated. We have restricted ourselves in Axiom 2.3 to a familiar linear learning model (Estes & Suppes, 1959). We begin with each word w of the given natural language having the value $d_1(w) = 1$. It is obvious from Axiom 2.3 that for all trials n, $0 \leq d_n(w) \leq 1$. For reasons we shall not expand upon here we chose as the value of the learning parameter $\theta = 0.03$. We show in Table 1 the denotational values of selected words from English, Chinese and German after approximately 400 learning trials using the corpus of commands described in Section 5. As desired the denotational value of the denoting words in the sense defined above remain close to 1, and the nondenoting words approach 0.

4.5. *Congruence of Nondenoting Words*

To reduce the number of grammatical rules and facilitate comparison of rules for different languages, we introduced in the last part of Axiom 2.5 the concept of comprehension congruence, applied here only to the limited case of nondenoting words. The sense of congruence used has, as the formulation of the axiom shows, a specific and definite content. The idea of congruence used here was developed earlier in Suppes (1973, 1991). To illustrate ideas, consider this German case:

Nimm die *Schraube!* $\sim (fa_1\ \$get\ (io\ (fo_1\ \$screw\ *)))$
Nimm den *Ring!* $\sim (fa_1\ \$get\ (io\ (fo_1\ \$washer\ *)))$
Nimm das *Loch!* $\sim (fa_1\ \$get\ (io\ (fo_1\ \$hole\ *)))$

Now generalize to the grammatical forms without denoting words, only category symbols, and we have:

$$A_1 \textit{ die OBJ } ! \sim (fa_1\ A_1\ (io\ (fo_1\ OBJ\ *)))$$
$$A_1 \textit{ den OBJ } ! \sim (fa_1\ A_1\ (io\ (fo_1\ OBJ\ *)))$$
$$A_1 \textit{ das OBJ } ! \sim (fa_1\ A_1\ (io\ (fo_1\ OBJ\ *)))$$

Then, using congruence, we may write as a single grammatical form

$$A_1\ [die]\ OBJ! \sim (fa_1\ A_1\ (io\ (fo_1\ OBJ\ *)))$$

where

$$[die] = (die,\ den,\ das).$$

We also note that in many cases, we include in the congruence class $[w]$ the empty string ϵ, if the corpus has generated such a variant grammatical form. For example, prior to computing congruence, we generated for Chinese the following three rules:

$$A \rightarrow \qquad O\ A_1 \sim (fa_1\ A_1\ O)$$
$$A \rightarrow \qquad ba\ O\ A_1 \sim (fa_1\ A_1\ O)$$
$$A \rightarrow xianzai\ ba\ O\ A_1 \sim (fa_1\ A_1\ O)$$

Computing congruence, we reduce these three rules to a single one:

$$A \rightarrow [ba]\ A_1\ O \sim (fa_1\ A_1\ O)$$

where $[ba] = (ba, xianzaiba, \epsilon)$.

In reporting the grammatical results for English, Chinese and German, we use an ordered triple notation for a given congruence class of nondenoting words of a rule common to all three languages: first the English words, then the Chinese words and finally the German words, with ϵ occurring as needed for each language. Thus the first common rule listed in Section 5 is

$$O \rightarrow [DA]\ S,$$

where $[DA] = (the; nage, zai, zhege; das, den, dem, der, die)$.

5. Grammars Generated

Comprehension grammars have been generated so far for English, Chinese, German, French, Dutch and Korean. Mainly because the first languages of the authors are English, German and Chinese respectively, we will concentrate on these three languages with remarks about the others. The corpus used has consisted of 456 commands, the last 60 of more complicated ones. In the first group commands range from the simple *Get the screw!* to *Place the screw on the plate!* In the last 60 a typical more complicated command is the following: *Put a small nut on the screw behind the washer left of the plate!*; *Ba yige xiaode luomu fang zai nage ban zuobian de dianquan houmian de nage luosiding shang!*; *Tu eine kleine Mutter auf die Schraube hinter dem Dichtungsring links von der Platte!*

Close translations of the 456 commands were used for the other languages. It was intended that the translations would be semantically equivalent but would also be faithful to idiomatic constructions in the given language.

The details of the learning curves that are generated from our procedure of machine learning are as described in the axioms but will not be analyzed here, because we already have a very thorough discussion of learning curves in our earlier paper (Suppes, Liang, & Böttner, 1992). We also do not analyze here in detail the learning of denotational value as characterized in Axiom 2.3. We do show in Table 1 the denotational values for some typical words, denoting and nondenoting in each of the three languages, English, Chinese and German, after approximately 400 learning trials for English and Chinese, and 800 for German. It is evident from the table that the denoting words have denotational values that are close to 1, and the nondenoting words values close to 0 as desired. It is important to emphasize that just as we assume a priori no grammar for any of the languages we study, we also do not assume in our machine learning an a priori classification of words as denoting or nondenoting.

We now turn to the three grammars generated for English, Chinese and German. The grammatical rules are written in context-free notation, in Table 2, with the congruence classes listed below.

Table 2. Comprehension grammars.

Chinese, English, German	
1. $O \rightarrow [DA]\ S$	
2. $O \rightarrow [IA]\ S$	
3. $S \rightarrow P\ S$	
4. $S \rightarrow OBJ$	
5. $G \rightarrow R\ [PO]\ O$	
6. $D \rightarrow R$	
7. $P \rightarrow P\ [\&]\ P'$	
8. $P \rightarrow [\neg]\ P$	
9. $P \rightarrow P\ [\vee]\ P'$	
Chinese, English	**English, German**
10. $A \rightarrow [ADV]\ A_4\ D\ O$	12. $A \rightarrow [ADV]\ A_1\ O\ [COP]$
11. $A \rightarrow A_4\ O$	13. $A \rightarrow [ADV]\ A_2\ G\ [COP]$
	14. $A \rightarrow G\ A_3\ O$
	15. $A \rightarrow A_3\ O\ [COP]\ G$
	16. $A \rightarrow [ADV]\ A_4\ [COP]\ O\ D$
	17. $S \rightarrow S\ [RP]\ P$
	18. $S \rightarrow S\ [RP]\ G$
Chinese	**German**
19. $A \rightarrow [ba]\ O\ A_1$	26. $A \rightarrow A_4\ [einen]\ S\ D\ [der]\ P\ ist$
20. $A \rightarrow [xianzai]\ G\ [na4]\ A_2$	27. $A \rightarrow A_4\ [nun]\ S\ D\ die\ G\ ist$
21. $A \rightarrow zai\ G\ A_3\ O$	28. $A \rightarrow A_4\ die\ S\ R\ von\ dem\ S\ D\ [der]\ G\ ist$
22. $A \rightarrow [ba]\ O\ A_3\ [dao1]\ G$	29. $A \rightarrow A_4\ die\ S\ R\ der\ S\ D\ die\ P\ ist$
23. $A \rightarrow [ba]\ O\ A_4\ D$	
24. $S \rightarrow G\ [de]\ S$	
25. $G \rightarrow O\ [de]\ R$	

The generation particularly uses Axiom 2.4 on grammar computations. Using the axiom on congruence (Axiom 2.5) to collapse rules that differ only in the occurrence of semantically equivalent nondenoting words, the number of rules for English is 18, for Chinese 18, and for

Table 3. Congruence classes.

$[DA]_1$	= (*the*;*nage*,*zai*,*zhege*;*das*,*dem*,*den*,*der*,*die*),
$[IA]_2$	= (*a*;*yige*;*eine*,*einem*,*einen*,*einer*),
$[PO]_5$	= (*of*,ϵ;ϵ;*von*,ϵ),
$[\&]_7$	= (*and*;*he*;*und*),
$[\neg]_8$	= (*not*;*busi*;*nicht*),
$[\vee]_9$	= (*or*;*huozhe*;*oder*)
$[ADV]_{10}$	= (*now*,*so*,ϵ;ϵ)
$[ADV]_{12}$	= (*now*,*so*,ϵ;*nun*,ϵ)
$[COP]_{12,13,15,16}$	= (ϵ;*ist*,ϵ)
$[ADV]_{13}$	= (*now*,*so*,ϵ;*dann*,*jetzt*,*nun*,ϵ)
$[ADV]_{16}$	= (*now*,*so*,ϵ;ϵ)
$[ADV']_{16}$	= (ϵ;*jetzt*,*nun*,ϵ)
$[RP]_{17}$	= (*that is*,*which is*;*der*,*die*,*ist die*)
$[RP]_{18}$	= (*that is*,*which is*,ϵ;*der*,*die*,*die sich*,ϵ)
$[ba]_{19}$	= (*ba*,*xianzai ba*,ϵ)
$[xianzai]_{20}$	= (*chao*,*name*,*xianzai*)
$[na4]_{20}$	= (*na4*,*na4li*)
$[ba]_{22}$	= (*ba*,ϵ)
$[dao1]_{22}$	= (*dao1*,*zai*,ϵ)
$[ba]_{23}$	= (*ba*,*xiazai ba*,ϵ)
$[de]_{24}$	= (*de*,*de nage*)
$[de]_{25}$	= (*de*,ϵ)
$[der]_{26}$	= (*der*,*die*)
$[einen]_{26}$	= (*einen*,*jetzt eine*)
$[ist]_{26}$	= (*ist*,ε)
$[nun]_{27}$	= (*die*,*nun die*)

German 20. What is of perhaps greater interest is the analysis of the structure of common rules in comparison with special rules for the particular languages. The basic numerical data are these. There are 9 rules that are common to English, Chinese and German where 'commonality' means that the category such as definite article now includes words from each of the three languages, in this particular case for example, *the*,*nage*,*zai*,*zhege*,*das*,*dem*,*den*,*der*, and *die*. In addition, there are 2 rules common to English and Chinese that are not in the German grammar, there are 7 rules common to English and German that are not used in Chinese, and no rules common only to Chinese and German.

What is remarkable about rules 1–9 in Table 2 is that none of them are high level rules for the generation of complete commands. The first 4 deal with the generation of object phrases, Rule 5 with the generation of a description of a region, Rule 6 with a direction, and the last 3 with properties. Note that the rules common to English and Chinese are high level rules for generating commands, and that 5 of the 7 rules common to English and German are high-level rules for generating commands. The remaining 2 are for generating object phrases. Five of the 7 rules in Table 2 special for Chinese are high-level rules for generating commands.

The big surprise is that no special grammatical rules are required for English. In no sense did we anticipate this outcome.

In Table 3 the congruence classes for the three languages are shown. The subscript on each class shows the grammatical rule of Table 2 on which it depends. For example, $[ADV]_{10}$ depends on Rule 10, which is relevant only for English and Chinese. The congruence class for Rule 1 is intuitively incorrect. The reason is simple to explain. The Chinese particles

xianzai,*dao1*,*zai*, and *ba* do not generally co-occur with object phrases, as the rules suggest, but with other categories of words. For example, *xianzai* has a meaning that is close to the English adverb *now*, the particle *dao1* occurs with verbs, as does *ba*. Here is an example from our corpus of the use of *zai*. *Get the screw on the plate!* has the following translation in Chinese:

Ba	*nage*	*zai*	*(nage)*	*ban*	*shang*	*de*	*luosiding*	*naqi!*
	the		(the)	plate	on		screw	get

In this example, we mention that the second occurrence of *nage* is deleted because it is awkward to say in Chinese *nage zai nage*. Note also in this example that although *ba* is connected with the verb *naqi*, *ba* comes at the beginning of the sentence, *naqi* at the end which would make for difficulties if we had to assign a specific semantic meaning to *ba*. What is important is that at the level of the commands we are considering, the nonintuitive Rule 1 is satisfactory for the purposes of comprehension. It would of course lead to very bad Chinese if applied to the production of utterances. We emphasize once again that in considering these Chinese examples, we are generating comprehension by uniform procedures that are the same across all languages. We are not claiming this can be done for production or even that for the ultimate reaches of comprehension it will be satisfactory. On the other hand it is important for machine learning purposes to understand how far uniform procedures can be satisfactorily exploited.

6. Problems Not Solved

Problems arise from the following assumptions: the association relation is one-one, and the internal language is a Lisp regimentation of English lexical semantics, as already mentioned. Here are some examples.

1. Contraction of denoting and nondenoting
   (a) German:
   Geh zur *Schraube!* $\sim (fa_1\ \$go\ (\$to\ (io\ (fo_1\ \$screw\ *))))$
   Geh zu *der Schraube!* $\sim (fa_1\ \$go\ (\$to\ (io\ (fo_1\ \$screw\ *))))$
   Geh zum *Loch!* $\sim (fa_1\ \$go\ (\$to\ (io\ (fo_1\ \$hole\ *))))$
   Geh zu *dem Loch!* $\sim (fa_1\ \$go\ (\$to\ (io\ (fo_1\ \$hole\ *))))$
   (b) French:
   Va au *trou!* $\sim (fa_1\ \$go\ (\$to\ (io\ (fo_1\ \$hole\ *))))$
   Va à *la vis!* $\sim (fa_1\ \$go\ (\$to\ (io\ (fo_1\ \$nut\ *))))$
2. Only one word for different internal symbols (French)
   Ferme *la porte!* $\sim (fa_1\ \$close\ (io\ (fo_1\ \$door\ *)))$
   Ferme *la porte!* $\sim (fa_1\ \$lock\ (io\ (fo_1\ \$door\ *)))$
3. More than one word corresponding to same denoting symbol
   (a) English:
   Put *the screw near the washer!*
   $\sim (fa_3\ \$put\ (io\ (fo_1\ \$screw\ *))\ (fr_1\ \$near\ (io\ (fo_1\ \$washer\ *))))$
   Place *the screw near the washer!*
   $\sim (fa_3\ \$put\ (io\ (fo_1\ \$screw\ *))\ (fr_1\ \$near\ (io\ (fo_1\ \$washer\ *))))$

(b) German:
Leg *die Schraube in die Schachtel!*
$\sim (fa_1\ \$put(\text{fo}\ \$screw*)\ fr_1\ \$in(io(\text{fo}\ \$box*))))$
Steck *die Schraube in das Loch!*
$\sim (fa_1\ \$put(\text{fo}\ \$screw*)(fr_1\ \$in(io(\text{fo}\ \$hole*))))$

(c) French:
vis ronde $\sim (fo_1\ \$round\ (fo_1\ \$screw*))$
douille cylindrique $\sim (fo_1\ \$round\ (fo_1\ \$washer*))$

4. Concepts that are expressed easily in one language can be more difficult to express in another language. For example, the English *Go behind the plate!* does not have a natural direct translation in Dutch, even though *go* can generally be translated by *gaan* in Dutch.

As a consequence, we made certain adjustments in the corpora of natural languages:

1. Since our system currently cannot identify different forms of the same denoting word, we mark the boundary between stem and ending by a hyphen, e.g., in French *vis rond-e* or German *dem rund-en Loch.*
2. There are no words in French and German that correspond to the English verb *pick up* with respect to its range of use. We chose the German translation *hochheben* because it resembles the English construction at least for the imperatives under consideration where the verb gets split into two parts and *heben* can be associated with *pick* and *hoch* can be associated with *up*. On the other hand, in cases like *Pick a screw!* the word *heben* does not appear to be usable and we switched to *nehmen.* In French we used *ramasser* as a translation for *pick up* and *saisir* as a translation for *pick.*
3. The English preposition *on* has no unique counterpart in German and French. Some contexts require it to be translated by *auf/sur* as in, e.g., *Put the screw on the plate!* for other contexts a translation by *an/à* would be more natural as in, e.g., *Put the nut on the screw!* We used *auf* and *sur* in all cases even in cases where the visual scene clearly suggests a different preposition.
4. If a language had no single word for a certain English expression but had only an expression which was longer than one word, we used this and marked it by underscoring in a fashion that the machine would take it for one word: for French we used *de_taille_moyenne* for the English adjective *medium.*
5. In French we used *à* instead of *au* since *à* is needed as a counterpart for a denoting symbol of the internal language. We also used *de le*, *de un*, and *de une* instead of *du*, *d'un*, and *d'une* respectively.

Our current research is focused on solving Problems 1 and 3 by generalizing Axiom 2.7 to permit several words of a given natural language to be associated to the same internal symbol.

7. Related Work

Our approach to language learning belongs to a semantically oriented paradigm of language learning but not to the well-known Gold-paradigm of syntactic language learning. For an

overview see Langley and Carbonell (1987). Learning systems using a semantic paradigm typically take pairs of sentences and representations of their meaning as input. Since most systems learn language from descriptive language the meanings are represented by pictures or simulations of pictures. This holds for the approaches by Anderson (1977, 1981), Siskind (1992) and Feldman and collaborators (Feldman, 1990; Regier, 1992; Stolcke, 1990). Our system acquires natural language from imperative rather than declarative language. Meanings are therefore represented by actions performed by the learner.

Our system does not require prior to learning any information that is specific to the language to be learned, in particular

- no initial set of word associations given before learning, as in Anderson (1977, 1981),
- no information about phrasal categories (whether as context-free grammar in the case of Siskind's DAVRA system or as "universal grammar" in the case of Siskind's other systems),
- no information about the distinction of whether a word is denoting or nondenoting.

Also unlike Siskind our learning proceeds strictly incrementally and errors in associations occur, as they do in human language learning.

Grammar induction typically has to solve the problems to classify words into categories, to establish constituents, and to classify recursively a potentially infinite set of strings. In order to derive constituent structures Anderson has a "graph deformation condition" which is similar to our axiom of form association in its effect and its function: it does the job of parsing a sentence by exploiting the properties of the meaning, i.e., given internal representation of the sentence. However, his formulation works for contiguous constituents only, which Anderson appears to think of as somehow to be more natural than noncontiguous constituents. This may not even be true for English and is certainly not true for the majority of languages with free word order.

Language learning systems may also differ with respect to their capacities: Our system and also Feldman's is restricted to comprehension. Anderson's (1977, 1981) and Siskind's (1992) systems serve the purposes of both comprehension and production.

Our work so far is focussed on the probabilistic acquisition of associations between natural language expressions and internal language expressions. As a consequence we have assumed that the learner has already acquired all the conceptual apparatus needed. This is too restrictive and we plan to weaken this assumption in our future research. Attempts to study the learning of concepts from visual experience have been undertaken by Siskind and Feldman and his collaborators. Siskind (1992) abstracts concepts of motion from sequences of state-descriptions invoking principles from the theory of perception. Feldman and collaborators make use of connectionist methods.

Acknowledgments

The research of Michael Böttner has been supported by grant 683/1-2 from the Deutsche Forschungsgemeinschaft. We are indebted to Michael Donio for the analysis of French and Paul Meyer for the analysis of Dutch.

References

Anderson, J.R. (1977). Induction of augmented transition networks. *Cognitive Science*, 1, 125–157.

Anderson, J.R. (1981). A theory of language acquisition based on general learning principles. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 97–103.

Estes, W.K., & Suppes, P. (1959). Foundations of linear models. In R.R. Bush, & W.K. Estes (Eds.), *Studies in mathematical learning theory*. Stanford, CA: Stanford University Press.

Feldman, J.A., Lakoff, G., Stolcke, A., & Hollbach Weber, S. (1990). Miniature language acquisition: A touchstone for cognitive science. Berkeley CA: ICSI.

Langley, P., & Carbonell, J.G. (1987). Language acquisition and machine learning. In B. MacWhinney (Ed.), *Mechanisms of language acquisition*. Hillsdale NJ: Erlbaum.

Regier, T. (1992). A connectionist architecture for learning motion-based spatial semantics. (Submitted to the 1992 Meeting of the Cognitive Science Society).

Siskind, J.M. (1992). Naive physics, event perception, lexical semantics, and language acquisition. M.I.T. Ph.D. Dissertation.

Stolcke, A. (1990). Learning feature-based semantics with simple recurrent networks. ICSI TR-90-015.

Suppes, P. (1973). Congruence of meaning. *Proceedings and Addresses of the American Philosophical Association*, 46, 21–38.

Suppes, P. (1991). *Language for humans and robots*. Oxford: Blackwell.

Suppes, P., Liang, L., & Böttner, M. (1992). Complexity issues in robotic machine learning of natural language. In L. Lam, & V. Naroditsky (Eds.), *Modeling complex phenomena*. New York: Springer.

Received April 10, 1992
Accepted July 6, 1994
Final Manuscript August 5, 1994