Robotica (1998) volume 16, pp. 425–431. © 1998 Cambridge University Press

Robotic machine learning of anaphora

Patrick Suppes and Michael Böttner

Ventura Hall, Stanford University, Stanford CA 94305-4115 (USA). E-mail: suppes@csli.stanford.edu.

(Received in Final Form: September 20, 1997)

SUMMARY

Our contribution tackles the problem of learning to understand anaphoric references in the context of robotic machine learning; e.g. *Get the large screw. Put it in the left hole*. Our solution assumes the probabilistic theory of learning spelt out in earlier publications. Associations are formed probabilistically between constituents of the verbal command and constituents of a presupposed internal representation. The theory is extended, as a first step, to anaphora by learning how to distinguish between incorrect surface depth and the correct tree-structure depth of the anaphoric references.

KEYWORDS: Anaphora; Machine learning; Robots.

Our focus is the problem of learning to understand anaphoric references in the context of robotic machine learning; e.g., *Get the large screw, Put it in the left hole.* Anaphoric references are an essential feature of natural language use.

> "Robust natural language man/machine communication requires a machine to have the ability to deal with anaphoric language in a perspicuous, transportable, non-*ad hoc* way."[1]

Although a huge amount of attention has been spent on anaphoric reference in the areas of philosophical logic and theoretical linguistics, see, e.g. references 2–5, we have found no reference with detailed learning models of anaphora but we may be unaware of some relevant prior work.

Our current work extends our probabilistic theory of learning spelt out in references 6–8. So far we have conducted two experiments. One was with a corpus of 460 English sentences in part of considerable complexity. The other was with 10 corpora of 400 sentences, each in a different language. Compared to most neural-net rates of learning, the learning was rapid. In each of the ten languages one cycle through the entire corpus was sufficient to produce a comprehension grammar that was intuitively correct.

The focus of our machine learning is restricted to comprehension. Most linguistic analysis is concerned with grammars detailed enough to produce natural utterances of the language being studied. A comprehension grammar, in contrast, as we characterize it here, can generate a superset of utterances. The rules are required only to lead to the correct semantic interpretation of an utterance of the language. Robots, like very young children, can easily have the capacity to understand language before they can produce it. Although it is difficult and subtle to collect accurate and anything like complete data on the comprehension grammar generated by very young children, the evidence is overwhelming that they comprehend much more than they can produce.

The learning robot is presented with pairs of a discourse together with a coerced action (simulated by some kind of internal representation). Associations are formed probabilistically between constituents of the verbal command and constituents of a presupposed internal representation. More specifically, we use anaphoric buffers where the robot's descriptions of objects are stored for a short period, together with some of their morphologic, syntactic, and discourse features.

The robotic framework and the associated corpora we test our program on are certainly restricted, although we have implemented our learning program on Robotworld, a standard robot used in academic settings for development purposes. However, our theory does not depend on the robotic framework but applies to many kinds of systematic language use, more or less in the sense of sublanguages.[9] Another sublanguage would be the language of physics word problems.[10,11]

The organization of this paper is as follows. In section 1 we describe our theory of machine learning of natural language. In section 2 we extend our theory to learning anaphora.

1. OUR THEORY WITHOUT ANAPHORA

For the purpose of simplification we have separated the task of learning the semantics of a language from learning its grammar, i.e. its lexicon and syntax. In our first approach our focus is on learning the expression assuming that the content is given. So we assume that the learner has already the notions of a screw, or a certain color, a spatial relation, or the action of putting some object at some place. What is learned are the English words for these notions and how they are put together to convey compound meanings.

1.1. Basic notions of our theory

Everything that is learned is stored in *memory.* Our memory consists of two parts: a working memory to hold its content for the time period of a single trial and a long-term memory to store associations of words, denotational values, associations of grammatical forms and memory traces.

The long-term memory is not empty at the beginning but

has stored in it an *internal language*. In the present study this internal language is stored in memory prior to learning and does not undergo any change during learning. Assume the robot is given the English command *Get a screw* together with the execution of this command. We here assume that the execution is given to the robot in some internal language. In Robotworld this language is RAIL. But RAIL is much too finegrained for our purpose. What we need is a language at the proper level of abstraction. In our experiments conducted so far we used English as our guideline. Our internal language was designed to have as many ultimate constituents as there are words denoting objects, properties, relations or actions in English. English words that do not denote anything of the afore-mentioned kind are called *nondenoting*—a distinction that will become important for the notion of a denotational value to be introduced below. In the example *Get a screw* the words *get* and *screw* are denoting and the word *a* is nondenoting. Consequently, our internal language will have to account for each of the denoting words by some symbol. We use *g* as a symbol for the action denoted by *get* and *s* as a symbol for the kind of objects denoted by *screw*. In addition we use * as a symbol for the set of objects that are in the perceptual environment of the learner. Symbols are combined to form compound expressions by semantic operations here referred to by *I*. The internal language expression for the command *Get a screw* will be $I(g, s, *)$.

Whatever gets into the memory gets there by *association*. We use this concept to establish the connection between linguistic expressions and their meanings. Here, association is a binary relation between commands, words and grammatical forms, on the one hand, and their counterparts in the robot's internal language, on the other hand.

By the *denotational value* of a word we understand the probability of that word to have a denotation. If this value is 1 the word is denoting and if it is 0 the word is nondenoting. Denoting words such as *nut* refer to elements of our categories; examples of nondenoting words are the definite and indefinite articles. Intuitively, only denoting words should acquire associations to elements of the environment, including possible actions, as represented internally. The purpose of this notion is to prevent nondenoting words from entering again and again into the probabilistic association procedure. We thereby exploit the fact that nondenoting words like *the*, *a*, and *is* have a higher frequency of occurrence and should be learned more easily than denoting words, which have less frequent occurrences. Consequently, we set the initial denoting value to be 1 for all words, for we assume no prior knowledge of which words are denoting in a given language.

Grammatical forms are generated by a principle of *generalization*. For example, the phrase *Get the nut* generalizes to the grammatical form A_1 *the OBJ*.

When a generalization is made, the particular word association on which it is based is stored with it in long-term memory, as the *memory trace* justifying the generalization. The memory trace maps associations of grammatical forms to sets of word associations. The memory trace of a grammatical form association thus keeps track of those word associations that gave rise to this particular grammatical form association. If one of the word associations in the trace of a grammatical form association is deleted, then so is this grammatical form association.

The concept of generalization is widely used in psychological theories of learning. The principle of association goes back at least to Aristotle, and certainly was used extensively by eighteenth-century philosophers like Hume long before psychology had become an experimental science. The fundamental role of association as a basis for conditioning is thoroughly recognized in modern neuroscience and is essential to the experimental study of the neuronal activity of a variety of animals. For similar reasons its role is just as central to the learning theory of neural networks, now rapidly developing in many different directions. Our distinction about kinds of memory is standard in psychological studies of human memory, but the details of our machine-learning process are not necessarily faithful to human learning of language, and we make no claim that they are. On the other hand, our basic processes of association, generalization, specification and rule-generation almost certainly have analogues in human learning, some better understood than others at the present time. In the general axioms formulated in this section we assume rather little about the specific language of the internal representation, although the examples that illustrate the axioms use the internal language described in the preceding section.

1.2. Background assumptions

We state informally, as background assumptions two essential aspects of any language learning device. First, how is the internal representation of an utterance heard, for example, for the first time, generated by the learner. Second, at the other end of the comprehension process, so to speak, is that of generating a semantic interpretation of a new utterance, but one that falls within the grammar and semantics already constructed by the learner.

Both of these processes ultimately require thorough formal analysis in any complete theory, but, as will become clear, this analysis is not necessary for the framework of this article. We give only a schematic formulation here.

a. Association by contiguity. When a learner is presented a sentence that it cannot interpret then it associates the utterance to patterns in its contiguous environment whose internal representation may, but not necessarily, be induced by its own free or coerced actions.

b. Comprehension-and-response axiom. If a learner is presented a sentence, then using the associations and grammatical rules stored in long-term memory, the learner attempts to construct a semantic interpretation of the sentence and respond accordingly.

1.3. Learning algorithm

For purposes of exposition we first describe the state when some English has successfully been learned. In a second step we describe how this state is arrived at.

If the learner has already learned as much English as to understand *Get a screw*, the learner's memory should contain (i) a semantically interpreted lexicon, (ii) a grammar of English, and (iii) a compositional semantics for that grammar. By a semantically interpreted lexicon we mean a list of English words together with the internal symbols they are associated to like this:

$$get\sim g,\ screw\sim s,\ \ldots$$

An English grammar would be a set of production rules like, e.g.

$$\text{(i) } A\rightarrow A_1\ O,\quad \text{(ii) } O\rightarrow a\ S,\quad \text{(iii) } S\rightarrow OBJ,$$

$$\text{(iv) } A_1\rightarrow get,\quad \text{(v) } OBJ\rightarrow screw,\ \ldots$$

The corresponding compositional semantics contains a list of associations of grammatical forms like this:

$$A_1\ O\sim I_1(A_1,\ O),\ a\ S\sim I_2(S),\ OBJ\sim I_3(OBJ,\ *),\ \ldots$$

An overview of the memory is given in Table 1. With this memory the learner will be able to interpret the English command *Get a screw* by deriving its translation $I(g, s, *)$ into its internal language. The derivation of the English command is standard like this:

$$A_1\ O\rightarrow A_1\ a\ S\rightarrow A_1\ a\ OBJ\rightarrow Get\ a\ screw$$

Corresponding replacement of English grammatical forms by their internal language counterparts returns:

$$I_1(A_1,\ O)=I_1(A_1,\ I_2(S))=I_1(A_1,\ I_2(I_3(OBJ,\ *)))$$
$$=I_1(g,\ I_2(I_3(s,\ *)))$$

In the following we shall describe how the memory can reach this state by learning from examples. We distinguish two cases: either no learning has occurred or some learning has occurred already. If no learning has occurred, the memory holds only the internal language part, i.e. left hand side of Table I.

Whenever a command is not understood correctly its execution is coerced. This coerced execution is represented by an internal language expression. A pair

$$Get\ a\ screw\sim I(g,\ s,\ *)$$

is formed from the natural command and the learner's internal language counterpart where ~ is the symbol used for association.

Table I. Memory with learned language stored

Internal language			English		
A	$\rightarrow$	$I_1(A_1,\ O)$	A	$\rightarrow$	$A_1\ O$
O	$\rightarrow$	$I_2(S)$	O	$\rightarrow$	$a\ S$
S	$\rightarrow$	$I_3(OBJ,\ *)$	S	$\rightarrow$	OBJ
OBG	$\rightarrow$	s	OBJ	$\rightarrow$	$screw$
A_1	$\rightarrow$	g	A_1	$\rightarrow$	get
	...			...	
			$screw$	~	s
			get	~	g
				...	
			$A_1\ O$	~	$I_1(A_1,\ O)$
			$a\ S$	~	$I_2(S)$
			OBJ	~	$I_3(OBJ,\ *)$

The learner probabilistically associates the words of the natural language command with the symbols of the internal language expression. There are three English words to associate to two internal symbols. We do not associate anything to $*$. Therefore there are six different ways to associate the words with internal symbols. The probability that the learner associates *screw* with s and *get* with g

$$screw\sim s,\ get\sim g \qquad (1)$$

is only 1/6. Let us assume this indeed happens.

By a principle of generalization the learner derives grammatical forms for both languages and derives the association

$$A_1\ a\ OBJ\sim I(A_1,\ OBJ,\ *).$$

The grammatical form will be stored in conjunction with those associations upon which the generalization was made. Since the function symbol I is an abbreviation for a structure that is internally complex, our memory holds the following form association

$$A_1\ a\ OBJ\sim I_1(A_1,\ I_2(I_3(OBJ,\ *))).$$

By principles of Form Association, Factorization and Filtering, this association will get broken down into smaller units like this:

$$A_1\ O\sim I_1(A_1,\ O),\ a\ S\sim I_2(S),\ OBJ\sim I_3(OBJ,\ *) \qquad (2)$$

By principles of rule generation, we get

$$A\rightarrow A_1\ O,\ O\rightarrow a\ S,\ S\rightarrow OBJ,\ OBJ\rightarrow screw,\ A_1\rightarrow get \qquad (3)$$

Taking (1), (2) and (3) together, our memory will contain exactly what was needed in memory to be able to execute the original command.

Consider now one of the cases with a wrong association hypothesized. An association that could arise with equal probability is

$$get\sim g,\ a\sim s$$

By the principle of generalization we would now arrive at the association of the following grammatical forms:

$$A_1\ OBJ\ screw\sim I(A_1,\ OBJ,\ *)$$

Assume the next trial would be the command *Get a nut*. The reaction would be wrong, since the learner would derive the same internal expression as was present in coercion for the command *Get a screw*. Therefore the reasonable next step would be to coerce

$$Get\ a\ nut\sim I(g,\ n,\ *)$$

By principle of association, the association $get\sim g$ would be kept but the association $a\sim s$ would be broken and the words a and *nut* would enter the sampling process each with an equal probability to be associated to n.

Our learning algorithm is specified by a set of axioms.The full set of axioms together with a detailed explanation of each axiom can be found in reference 8.

2. ANAPHORIC EXTENSION

The standard view is that anaphora, which literally means back-reference, is a function of pronouns. Pronouns can be

used either deictically or anaphorically. The Greek word δειξισ means pointing. Pronouns used deictically refer to what can be pointed at in the situation of the utterance taking place, which is what is present for speaker and hearer in their common perceptual environment. Pronouns used anaphorically refer to what is not present in the speaker's and hearer's perceptual environment but what may be expected to be present in the memory common to speaker and hearer. Although the anaphoric function is closely tied to this category, pronouns are not the only means to express an anaphoric relationship. Another means of expressing anaphoric relationships is the definite article like in, e.g. *I met a strange couple. The man carried a big suitcase.*

Pronouns come in various subcategories like, e.g., personal pronouns, reflexive pronouns, possessive pronouns, or demonstrative pronouns. Although literally a pronoun is anything that can replace a noun, pronouns occur in a variety of syntactical categories. Personal pronouns like *he* or *you* occur as noun phrases. Possessive pronouns like, e.g. *my* and demonstrative pronouns like, e.g. *this* occur as adjectives or, rather, determiners. And there are pronouns that can occur in an adverbial position like, e.g. *there.*

Standard examples occurring in a robotic context are

Go to screw. Pick it up.

Get a screw. Put it in a box.

Get a box. Put a screw in it.

Pick up a red screw and a black screw. Put the larger one in a box.

These examples have in common that the anaphoric expression refers to a physical object. In the example

Put a black screw near the hole. Put a nut of the same colour near the plate

the adjective *same* refers anaphorically to *black.* It is also possible to refer to actions as illustrated by the following example:

Put a nut on the red screw. Do the same on a black screw.

Here the phrase *do the same* refers back to the antecedent *put a nut on.*

Gender often distinguishes anaphoric reference, as in:

Susan and Bill went to the movies. She did not like it.

But most of the commands to robots we have in mind are about inanimate physical objects, for which, in English, the proper anaphoric pronoun is uniformly *it.*

Physical implausibility can sometimes disambiguate anaphoric reference in robotic contexts:

Put the nut in the box. Then put the lid on it.

Nuts do not have lids ordinarily, so *box* is the anaphoric reference. The example is intuitively simple, but this general kind of feature is not, for it implies the robot needs to have a great deal of commonsense knowledge about the world, with useful limitations not easily set. In spite of its importance we do not pursue further the learning of criteria of physical plausibility. The feature whose learning we do study is that of grammatical depth, one of the most important for succcessful anaphoric disambiguation.

Because of the partial and preliminary character of our results we sketch this anaphoric extension of our theory informally. We assume, first, that by the methods of association described above, but extended to include an internal buffer B, the word *it* is associated to B. But B holds in the internal representation the necessarily temporary reference by association of *it.* From this point on, we assume the internal language is Lisp. How is this selected anaphoric reference learned, in the restricted case we are considering? Scanning backward in the discourse, various measures of newness or closeness can easily be defined. For simplicity we restrict ourselves to two. The measure $s(w)$ is the number of words between *it* and the first word of the phrase w describing an object. The measure $d(w)$ is the measure of depth of the Lisp expression that corresponds to w.

Our problem is to formulate a model for learning which of these two object measures of "depth" is better for correctly assigning anaphoric reference. This can be done in many ways. We choose an approach similar to the one used for denotational value, but with certain differences as well.

Let $w_{1,n}$ be the weight of measure d on learning trial n, and let $w_{2,n}$ be the weight of learning measure s on trial n. We set $w_{1,1}=w_{2,1}=1/2$, i.e. at the beginning of trial 1 both weights are 1/2. The decision rule is to use either the minimum $d_n=\delta_n$ or the minimum $s_n=\sigma_n$ on trial n. The probability of using δ_n is $w_{1,n}$, and, of using σ_n, is, of course, $w_{2,n}$.

The learning model for the weights is a linear one like that used for denotational value. On each trial n, if δ is the distance of the correct anaphoric reference and σ is not, then

$$w_{1,n+1}=(1-\theta)w_{1,n}+\theta$$

$$w_{2,n+1}=(1-\theta)w_{2,n},$$

where $0<\theta\leq 1$, with θ the learning parameter. The learning equations are interchanged when σ_n is correct and δ_n is not. In the denotational computations we found $\theta=0.03$ worked well. In case both δ and σ choose the correct, or, jointly, the wrong reference, the weights remain unchanged:

$$w_{i,n+1}=w_{i,n} \quad \text{for} \quad i=1, 2.$$

Consider some examples.

Go to the screw. Pick it up.

Here δ (*the screw*) and σ (*the screw*) are both correct, as the only possibility. So for this learning trial there would be no change in the weights.

In contrast, consider

Go to a screw left of the box. Pick it up.

Here

d(*a screw left of the box*)=2 d(*the box*)=5

s(*a screw left of the box*)=1 s(*the box*)=1

So δ picks out *a screw left of the box* and σ picks out *the box*

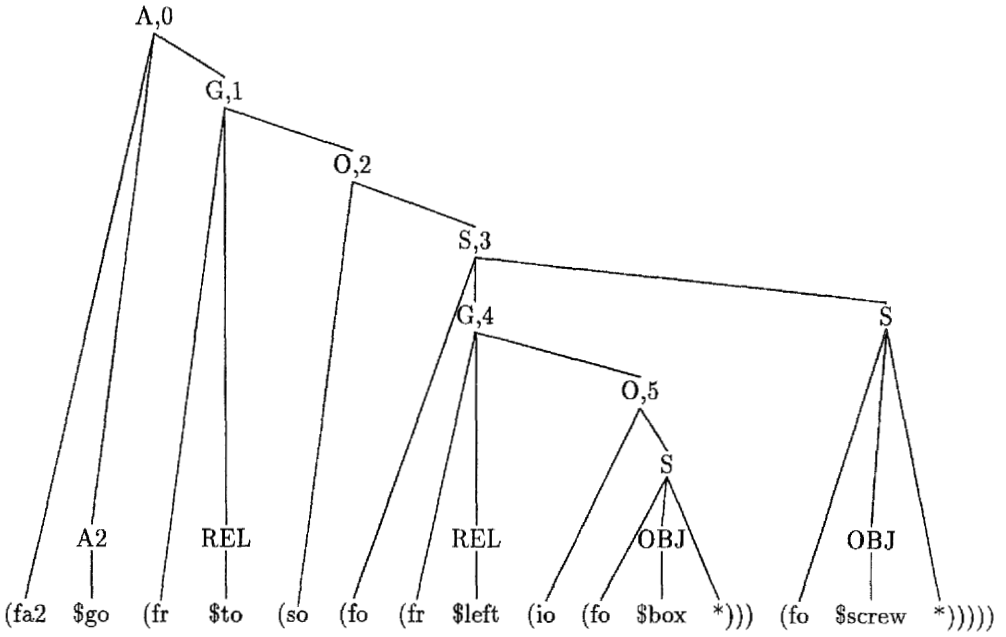

Fig. 1. Lisp tree for *Go to a screw left of a box*. The Lisp function *fa*2 turns an action of subcategory *A*2 and a region G into an action *A*; the Lisp function *fr* turns a spatial relation *REL* and an object *O* into a region *G*; the semantic operation *so* selects an object *O* from a set *S* of objects; the semantic operation *io* identifies an object out of a set *S* of objects; the semantic operation *fo* selects a set *S* of objects of category *OBJ*.

as correct anaphoric reference. Since δ is correct, the weight w_1 is increased, and the weight of w_2 decreased.

We spell out here in more detail the computation of *d*. We compute the depth of an antecedent of a pronoun by counting the nodes from the root of the Lisp tree assuming the root node to be 0. So in the Lisp tree in Figure 1 the object computed by the Lisp expression corresponding to the English expression *the box*, which is

(io (fo $box *)),

gets count number 5, and the object computed by the Lisp expression corresponding to *a screw left of the box*, which is

(so (fo (fr $left (io (fo $box *)))) (fo $screw *)),

gets count number 2.

We ran our learning model on a small test corpus of the following eight sentences.

1. *Pick up a screw to the right of the washer. Put it in a box.*
2. *Pick up the screw in the yellow box. Put it in the green box.*
3. *Put in the red box a red screw. Pick it up.*

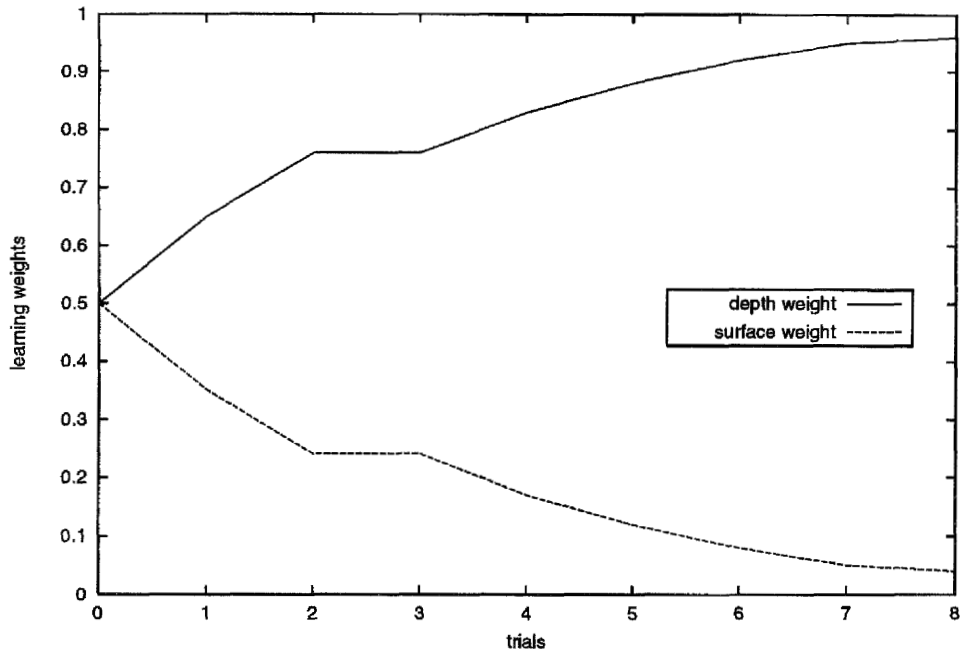

Fig. 2. Learning curves for depth and surface weights.

4. *Put a screw in the hole of the plate. Now put a nut on it.*
5. *Go to the nut left of the screw. Pick it up.*
6. *Put a bolt in the red hole which is next to the black box. Then put a washer on it.*
7. *Put a red nut on a small screw. Next put it in the green hole.*
8. *Get a small washer that is on the red plate. Put it on a bolt.*

In the following we list the d and s values for respective phrases that could be considered as possible antecedents of the anaphoric pronoun *it* in the eight robotic commands listed above. The numbering of the phrases below corresponds to the commands above.

1. *a screw to the right of the washer*: $d=1$, $s=8$
   the washer: $d=4$, $s=2$
2. *the yellow box*: $d=4$, $s=3$
   the screw in the yellow box: $d=1$, $s=6$
3. *a red screw*: $d=1$, $s=3$
   the red box: $d=2$, $s=6$
4. *a screw*: $d=1$, $s=12$
   the plate: $d=5$, $s=6$
   the hole of the plate: $d=2$, $s=9$
5. *the screw*: $d=4$, $s=2$
   the nut left of the screw: $d=2$, $s=6$
6. *a bolt*: $d=1$, $s=17$
   the black box: $d=4$, $s=7$,
   the red hole which is next to the black box: $d=2$, $s=14$
7. *a small screw*: $d=2$, $s=4$
   a red nut: $d=1$, $s=8$
8. *a small washer that is on the red plate*: $d=1$, $s=9$
   the red plate: $d=6$, $s=3$

Learning curves for depth and surface weights $w_{1,n}$ and $w_{2,n}$ are shown in Figure 2. We note that d is the correct measure in all of the commands considered and the measure s is correct only once, on trial 3.

This method with two measures can be extended to other essential features of learning anaphora. The additional features we immediately have in mind are physical consistency and gender.

In English, but not in inflected languages like German, we can handle most robotic commands without gender. So the anaphoric extension described here can easily be implemented in a natural way as an extension of our earlier work with Robotworld.

References

1. Bonnie Lynn Webber, *A Formal Approach to Discourse Anaphora* (Garland, New York, 1979).
2. Richard M. Smaby, "Ambiguous coreference with quantifiers" **In:** (Fritz Guenthner and S. J. Schmidt, editors) *Formal Semantics and Pragmatics for Natural Languages* (Reidel, Dordrecht, 1978) pp. 37–75.
3. Tanya Reinhart, *Anaphora and Semantic Interpretation* (Croom Helm, London, 1983).
4. Hans Kamp "A theory of truth and semantic representation" **In:** (Jeroen Groenendijk, Theo Janssen and Martin Stokhof, editors) *Formal Methods in the Study of Language.* Mathematical Centre Tracts (Mathematisch Centrum, Amsterdam, 1981) pp. 277–322.
5. Hans Kamp and Uwe Reyle *From Discourse to Logic* (Kluwer, Dordrecht, 1993).
6. Patrick Suppes, Lin Liang and Michael Böttner "Complexity issues in robotic machine learning of natural language" **In:** (Lui Lam and Vladimir Naroditsky, editors) *Modeling Complexity Phenomena* (Springer, Berlin, 1992) pp. 102–127.
7. Patrick Suppes, Michael Böttner and Lin Liang, "Comprehension grammars generated from machine learning of natural languages" *Machine Learning* **19,** 133–152 (1995).
8. Patrick Suppes, Michael Böttner and Lin Liang, "Machine learning comprehension grammars for ten languages" *Computational Linguistics* **22**(3), 329–350 (1996).
9. Richard Kittredge and John Lehrberger, *Sublanguage. Studies of Language in Restricted Semantic Domains* (De Gruyter, Berlin, 1982).
10. Patrick Suppes, Michael Böttner, Lin Liang and Ray Ravaglia, "Machine learning of natural language: Problems and prospects" **In:** (Michael de Glas and Z. Pawlak, editors) *WOFCAI 95. Proceedings of the Second World Conference on the Fundamentals of Artificial Intelligence,* Angkor, Paris (1995) pp. 511–525.
11. Patrick Suppes, Michael Böttner and Lin Liang, "Machine learning of physics world problems: A preliminary report" **In:** (Atocha Aliseda, Rob Van Glabbeek and Dag Westerstahl, editors) *Computing Natural Language,* pages 141–154 (CSLI Press, Stanford, 1998) pp. 141–154.

APPENDIX
AXIOMS OF LEARNING

1. Computations using Working Memory

1.1. **Probabilistic Association.** On any trial, let s be associated to σ, let a be in the set of words of s not associated to any internal expression of σ, and let A be the set of expressions of the internal language made available for association and let α be in A but not currently associated with any word of s. Then pairs (a, α), are sampled, possibly using the current denotational value, and associated, i.e. $a \sim \alpha$.

1.2. **Form Generalization.** If $g(g_i') \sim \gamma(\gamma_i')$, $g_i' \sim \gamma_i'$, and γ' is derivable from X, then $g(X_i) \sim \gamma(X_i)$, where i is the index of occurrence.

1.3. **Grammar—Rule Generation.** If $g \sim \gamma$ and γ is derivable from X, then $X \rightarrow g$.

1.4. **Form Association.** If $g(g') \sim \gamma(\gamma')$ and g' and γ' have the corresponding indexed categories, then $g' \sim \gamma'$.

1.5. **Form Specification.** If $g(X_i) \sim \gamma(X_i)$, $g' \sim \gamma'$, and γ is derivable from X, then $g(g_i') \sim \gamma(\gamma_i')$.

1.6. **Content Deletion.** The content of working memory is deleted at the end of each trial.

2. Changes in State of Long-term Memory

2.1. **Denotational Value Computation.** If at the end of trial n a word a in the presented verbal stimulus is associated with some internal expression α, then $d(a)$, the denotational value of a increases and if a is not so associated $d(a)$ decreases. Moreover, if a word a does not occur on a trial, then $d(a)$ stays the same unless the association of a to an internal expression α is broken on the trial, in which case $d(a)$ decreases.

2.2. **Form Factorization.** If $g \sim \gamma$ and g' is a substring of g that is already in long-term memory and g' and γ' are

derivable from X, then g and γ are reduced to $g(X)$ and $\gamma(X)$. Also $g(X) \sim \gamma(X)$ is stored in long-term memory, as is the corresponding grammatical rule generated by Axiom 1.4.

2.3. **Form Filtering.** Associations and grammatical rules are removed from long-term memory at any time if they can be generated.

2.4. **Congruence Computation.** If w is a substring of g, w' is a substring of g' and they are such that
(a) $g \sim \gamma$ and $g' \sim \gamma$,
(b) g' differs from g only in the occurrence of w' in place of w,
(c) w and w' contain no words of high denotational value,
then $w' \approx w$ and the congruence is stored in long-term memory.

2.5. **Formation of Memory Trace.** The first time a form generalization, grammatical rule or congruence is formed, the word associations on which the generalization, grammatical rule or congruence is based are stored with it in long-term memory.

2.6. **Deletion of Associations.**
(a) When a word in a sentence is given a new association, any prior association of that word is deleted from long-term memory.
(b) If $a \sim \alpha$ at the beginning of a trial, a appears in the utterance s given on that trial but α does not appear in the internal representation σ of s, then the association $a \sim \alpha$ is deleted from long-term memory.
(c) If (i) no internal representation is generated from the occurrence of a sentence s, (ii) σ is then given as the correct internal representation, and (iii) there are several words in s associated to an internal expression α of σ such that the number of occurrences of these words is greater than the number of occurrences of α in σ, then these associations are deleted.

2.7. **Deletion of 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.