A Robot That Uses Existing Vocabulary to Infer Non-Visual Word Meanings from Observation

Kevin Gold and **Brian Scassellati**
Department of Computer Science
Yale University
New Haven, CT, USA
kevin.gold@yale.edu and scaz@cs.yale.edu

Abstract

The authors present TWIG, a visually grounded word-learning system that uses its existing knowledge of vocabulary, grammar, and action schemas to help it learn the meanings of new words from its environment. Most systems built to learn word meanings from sensory data focus on the "base case" of learning words when the robot knows nothing, and do not incorporate grammatical knowledge to aid the process of inferring meaning. The present study shows how using existing language knowledge can aid the word-learning process in three ways. First, partial parses of sentences can focus the robot's attention on the correct item or relation in the environment. Second, grammatical inference can suggest whether a new word refers to a unary or binary relation. Third, the robot's existing predicate schemas can suggest possibilities for a new predicate. The authors demonstrate that TWIG can use its understanding of the phrase "got the ball" while watching a game of catch to learn that "I" refers to the speaker, "you" refers to the addressee, and the names refer to particular people. The robot then uses these new words to learn that "am" and "are" refer to the identity relation.

Introduction

Anyone who has ever attempted to watch an untranslated foreign film knows that trying to learn a language from scratch is hard. Even when a speaker is pointing and looking at something, the meaning can be ambiguous; adapting an example from (Quine 1960), a speaker pointing at a rabbit could be saying, "That belongs to my friend," or, "Those things keep eating my flowers," or even, "I wonder if having four rabbit's feet is luckier than having just one."

Watching a film in a language with which one has some acquaintance is a different matter, because one can leverage existing knowledge of grammar and vocabulary to learn new words. This is particularly true when no more than one word in a sentence is new to the learner. Using the previous example, if a young child heard our rabbit-pointer say, "That belongs to my *compatriot*," the child could remember the word and match it to the speaker's friend when he came to collect the pet. Hearing the speaker say, "Those keep eating my *hyacinths*," a child could infer that a hyacinth is some kind of plant.

So far, much of the research in robotic word-learning has concentrated on the case in which the robot knows nothing about the language except its raw sounds, or phonemes. The most notable recent examples were the Roy's CELL system (Roy & Pentland 2002), and Yu's eyetracking language system (Yu & Ballard 2004). Impressively, these systems did produce reliable associations between phoneme sequences and visual stimuli, despite the complexities of learning with real audio and visual data. However, neither of these included a way to leverage this new vocabulary in order to learn new words, nor could they produce grammatical utterances from what they had learned. The Neural Theory of Language Project at Berkeley also produced notable word learning projects (Regier 1996; Bailey 1997), but emphasized how particular words could be learned in isolation, rather than in a linguistic context. By comparison, text-based natural language processing has long used approaches that integrate semantics and syntax, but there has been a surprising dearth of research that applies formal semantics to the situation of a robot with noisy sensors attempting to learn words through observation.

This paper describes TWIG, a word-learning system that can use its knowledge of meaning and grammar to help it learn new words. TWIG parses each sentence to the best of its ability in Prolog, grounding all of the terms it understands in predicates generated by the robot's sensory modules. When TWIG encounters a sentence that cannot be parsed or grounded using its current vocabulary, it attempts to infer who or what the speaker is talking about. Over time, it uses the weight of statistical evidence to hypothesize a more general word meaning. The system is not meant to occupy the same niche as systems that learn first words, but instead builds on such systems by adding more linguistic structure.

Below, we present the details of the system and the results of two experiments using the vision system of our lab's robot, Nico. In the first experiment, Nico uses TWIG to learn the meanings of "I," "you," and some proper names from watching two people pass a ball back and forth, saying "I got the ball" or "you got the ball." In the second experiment, Nico uses the words it learned in the first experiment to learn the meanings of "am" and "are" from the sentences "I am (name)" and "You are (name)." These experiments demonstrate that the system excels at learning words that

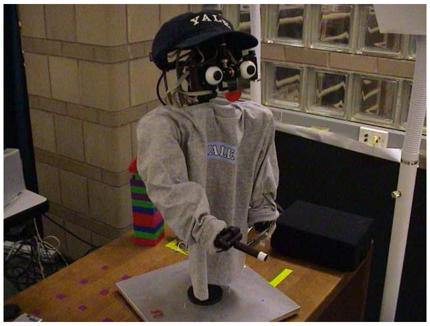

Figure 1: The robot Nico, on which the model was implemented.

could not be learned using previous techniques, such as deictic pronouns and linking verbs, and that the system does in fact make use of its acquired knowledge to learn more words. By the end of the second experiment, the system can parse sentences composed completely of words it did not know before the experiments were performed.

Though Gold and Scassellati previously presented work on using chi-square tests to learn “I” and “you” (Gold & Scassellati 2006), that work was not built around a framework of predicate logic, did not make use of grammatical information, and was not truly scalable. A framework of formal semantics allows TWIG to learn transitive and even linking verbs, allows the system to seamlessly use its new semantic knowledge for learning more words, and opens up the possibility of integration with larger knowledge bases and traditional natural language processing methods.

Robotic Implementation

We here describe the vision and auditory systems of the robot Nico (Figure 1), so as to make the operations of the TWIG back end more concrete. Nico is an upper-torso humanoid robot with the head and arm kinematics of a one-year-old child.

Visual Processing

The experiments performed here used one of Nico’s two wide-angle CCD cameras, which produced 320×240 pixel images at 30 frames per second. Subsequent vision processing was performed by three modules running on three separate Pentium 4 processors running the QNX Neutrino real-time operating system.

Two of the modules were devoted to face detection and head pose classification. These ran the face-detection algorithm of (Viola & Jones 2004) at 10 fps, using the Intel OpenCV implementation. One module used the OpenCV module trained to find profile faces, while the other used the classifier for faces oriented toward the camera.

Using some empirically measured conditional probabilities, we constructed a Hidden Markov Model in which the hidden state was the subject’s actual facing direction, and the evidence at each time step was the detection/non-detection output of the two face detectors. Different face detections over time or from different detectors were incorporated into the same HMM estimation if their areas overlapped; otherwise, a new HMM was created. The hidden state of actual facing direction for each HMM was calculated in real-time.

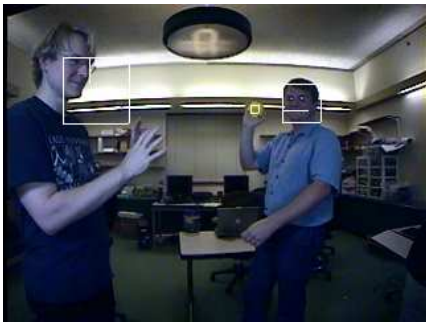

Figure 2: The visual processing step finds faces, their orientation (indicated by the small vertical or horizontal lines), and the ball (center-right).

The yellow ball of Legos mentioned in the experiments was found using a simple filter for its color. Figure 2 shows an image from Nico’s cameras that has been annotated with the results of the visual processing modules. Despite the smoothing performed by the HMM, the facing results were still somewhat noisy, as we shall describe in the results.

Auditory processing

Audio was collected using two microphones placed roughly 25 cm apart on a tray 75 cm away from the robot’s camera. The robot judged speech to come from the left or right by comparing the volume of input to the two microphones over time.

The system used the Sphinx-4 speech recognition system to parse audio into words. A simple context-free grammar incorporating all of the words used in our experiments was used to create the system’s language model. Though recognition was fairly accurate for our small CFG, the fact that Sphinx did not accurately report when utterances began and ended resulted in many errors of synchronization between speech and visual processing.

Predicates

The input from the robot’s sensory systems was converted into the following symbols and logical predicates before being passed to the TWIG system.

Symbols were created for each face, and also for the ball; below, we shall refer to these symbols as l and r for the person on the left and right, respectively, and b for the ball. The system also used the symbol n for itself. Each face and the ball received a predicate that uniquely identified it; we shall refer to these as $lprop(X)$, $rprop(X)$, and $ball(X)$. If the ball was within a threshold distance of a face, the predicate $has(P, b)$ is true, where P was the symbol for that person.

On detecting speech, the audio system produced the predicate $tells(X, Y, Z)$, where X was the speaker, Y was the person being addressed, and Z was the word segmentation produced by Sphinx. The person being addressed was inferred to be either the other face, if the speaker was viewed in profile, or the robot itself if the speaker was looking toward the camera.

The system also had access to the identity predicate; $ident(X, X)$ was true for all objects X.

The TWIG system

The two halves of the TWIG system correspond to the two kinds of meaning that are dealt with in formal semantics: *extension* and *intension*. The *extension* of a word is the thing in the world that it "points" to; it is sometimes also called the "referent." The extension of a word is its meaning in a specific context. The *intension* of a word, on the other hand, is its more general meaning: the conditions under which the word can correctly be applied. This distinction can be traced back to the philosopher Frege, and was greatly expanded upon in the work of Richard Montague (1974).

TWIG stands for "Transportable Word Intension Generator," because its final output is a hypothesis for the intension of a word. It is *transportable* because it rests on logical predicates as a layer of abstraction above a robot's particular sensory system, but its statistical approach makes it resilient to the noise and variability that robotic sensors inevitably produce.

The system finds the extension of a word through logical reasoning, but finds the intension through statistical reasoning. These two steps shall now be described in greater detail.

Parsing and Finding the Extension

The TWIG system adapts the following discrete-clause grammar from (Pereira & Shieber 1987):

```
s(S, W) --> np(VP^S, W), vp(VP, W).
np((E^S)^S, W) --> pn(E, W).
np(NP, W) --> det(N^NP, W), n(N).
vp(X^S, W) --> tv(X^IV), np(IV^S, W).
vp(IV, _W) --> iv(IV).
```

The abbreviations on phrase types are typical: np for "noun phrase," iv for "intransitive verb," and so on. pn covers both proper nouns and pronouns. W is a pointer to a list of predicates indicating the state of the world, which must be passed as an argument so that words and phrases can be grounded in the world state as they are parsed.

A term in the form $X\,\hat{}\,\Phi$ is shorthand for the lambda expression $\lambda X.\Phi$, the notation for a function Φ with an argument X. In Montague's semantics, sometimes now simply called "formal semantics" (Saeed 2003), the meanings of words and phrases could be expressed using lambda notation over logical expressions: the verb "has," for instance, could be expressed as $\lambda X.\lambda Y.possesses(X, Y)$, indicating that "has" refers to a function $possesses(X, Y)$ that takes two arguments, a possessor and possessed. In the Prolog language, these terms can be used inline in discrete-clause grammars, and the arguments of the functions are substituted as the parse provides them (see Figure 3).

In the case of verbs and nouns, words at the lowest level are associated with their lambda calculus definitions:

```
tv(Word, X^Y^pred([Word, X, Y])).
iv(Word, X^pred([Word, X])).
n(Word, X^pred([Word, X])).
```

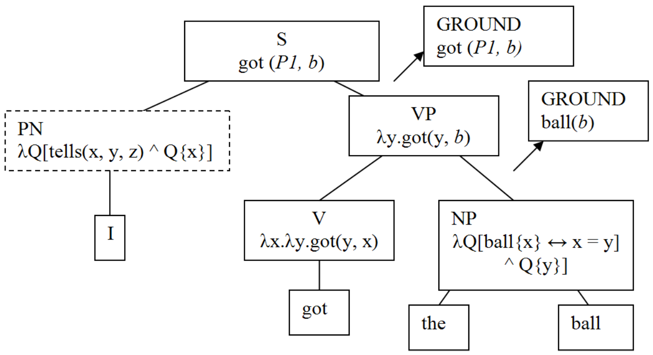

Figure 3: Parsing a sentence with an undefined word, ”I.” The parse partially succeeds on the right, and the system finds that the whole sentence can be grounded if "I" refers to person *P1*, who has the ball. The missing definition, that "I" refers to whoever is speaking, can only be learned over time.

During parsing, these expressions simply create logical forms with the same names as the corresponding words, and the correct number of arguments: one for intransitive verbs and nouns, two for transitive verbs. The predicate `pred([P, ...])` represents the predicate $P(...)$ in the robot's sensory representation; we shall see below that it is useful to treat the predicate P as a variable.

Proper nouns, pronouns, and noun phrases beginning with "the" are immediately grounded in the robot's environment. In Prolog, this is expressed as follows:

```
det(the,W,(X^S)^S) :- contains(W,S).
pn(PN,W,X) :- contains(W,pred([PN,X])).
```

The predicate `contains(W, X)` is true if the world W entails the fact X. If an object to which the word or phrase applies is found, its symbol takes the place of the corresponding predicate. For instance, on parsing "the ball," the system searches the world W for a symbol X such that $ball(X)$. If $ball(b)$ is found in W, $X\,\hat{}\,ball(X)$ is replaced with b.

If there is no object in W that matches a word that must be grounded, the parse fails. Then the system is allowed to guess one word extension that it does not actually know. An unconstrained variable A is appended to the world W before passing it into the parser, and the parser solves for A. This effectively allows the robot to hypothesize a fact of the form $Word(Object)$, where $Word$ is a predicate named after the new word and $Object$ is the object to which it refers.

For example, suppose the robot hears the statement "Alice got the ball." It does not know who Alice is, but it sees girl a holding a ball b and girl e holding nothing. The parse fails the first time because the robot does not know Alice's name. It does, however, know that "got the ball" parses to $\lambda X.has(X, b)$. On retrying with the free variable, the robot finds that hypothesizing $Alice(a)$ allows it to match the sentence to $has(a, b)$, a fact it already knows. Thus, "Alice" is assumed to refer to a: the system has successfully inferred the extension.

But the system does not assume that the word “Alice” can *only* refer to a – it could be some kind of pronoun, or refer to some other property that Alice has. In the next step, the system begins to accumulate statistical evidence for the intension of the new word.

Finding the Intension

Once an extension for a word has been found, the system next forms hypotheses about the intension of the word. The system searches its knowledge about the world W for all facts about the extension. This includes single-argument predicates as well as relations: for example, the facts retrieved if the ball b were the extension might include both $ball(b)$ and $got(a, b)$.

A new definition, or intension, for a word in the TWIG system consists of two parts: a predicate P and an argument number i. We define the following semantics for the `define` operator:

$$\text{define}(w, P, i) \Longleftrightarrow P(\underbrace{\ldots}_{i-1}, o, \ldots) \models w(o) \tag{1}$$

We sometimes use the shorthand $[[w]] = P@i$, which is equivalent to define(w, P, i). (The bracket notation is adapted from (Dowty, Wall, & Peters 1981).) In the case of single-place predicates, this intuitively allows a word to be defined by an already existing single-place predicate: for example, $[[\text{ball}]] = ball@1$. In the case of predicates of higher arity, this allows us to define words in terms of an object or person’s relation to something else. For example, given a predicate $tells(X, Y, Z)$ that holds if X is speaking to Y and saying Z, we can define $[[\text{I}]] = tells@1$ and $[[\text{you}]] = tells@2$, corresponding to the notion that “I” is the speaker and “you” is the person being addressed.

The TWIG system generates a list of such possible definitions every time a new word is associated with an extension, based on all facts that hold about the object. Many of these predicates will be fairly uninformative – for instance, the identity will always hold for every object with itself. The system thus can’t simply count the number of times a definition has held for a word; it needs a way to find the predicates about which the word is the most informative.

For this reason, the TWIG system uses chi-square tests to find the most statistically significant associations between words and definitions. Pairwise chi-square tests have been used in the past to find words that appear together in text more often than one would expect due to chance (Manning & Schütze 1999); here, we use them to find word-definition pairs that have held more often than chance would dictate.

For each possible definition Φ_{ip}, corresponding to predicate argument i and predicate p, the system counts the number of times ϕ_{ip} that any word’s extension has fit the definition. For each word W_j, the system counts the number of times w_j the word has been used, and the number of times it has been used for each predicate-place pair, w_{ijp}. In addition, the system tracks the total number of words σ that have referred to extensions so far. Using these quantities, it is straightforward to show that the system can compute chi-square values for each word-definition pair.

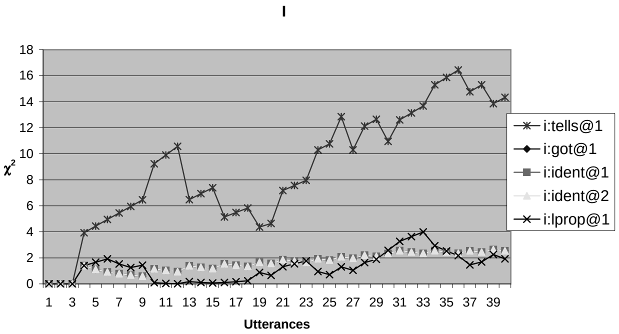

Figure 4: Sample chi-square values over time for one subject (M.D.) in experiment 1, for the word “I.” The correct definition is tells@1, corresponding to the speaker.

Chi-square values may be high for word-definition pairs for which the word appears *less* often than expected, which is not generally helpful for a word definition. Thus, we exclude the cases where $w_{ijp} < \mathbf{E}[w_{ijp}] = w_j \phi_{ip} / \sigma$. Otherwise, the system estimates the best intension for a word to be the definition with the highest chi-square value of all that word’s definition pairs.

Assuming that the arity of the largest predicate is a constant, the memory required for the system is $O(PV)$, where P is the number of different predicates in the knowledge base and V is the size of the vocabulary to be learned. In practice, the storage requirements can be much smaller, because word-definition pairs that are never observed do not need to be stored.

If the program is halted, the chi-square values themselves are not sufficient to resume learning at a later time, but the counts are. On starting again, TWIG reads a data file to obtain all the information it learned previously, and asserts `define`(w, p, i) for any word-definition pair that is higher than all other chi-square values for the same word, and also exceeds a threshold of significance of $p < 0.05$ ($\chi^2 > 3.84$). This definition can then be used for parsing sentences normally or making inferences about other words.

Learning transitive verbs

The explanations above focused on the case of words that are interpretable as single-place predicates, such as nouns, pronouns, and intransitive verbs. Transitive verbs are learned in almost the same way. On encountering a new transitive verb v, the system’s parse fails the first time, and the free variable A is appended to the world description W. On the second pass, if the subject and object noun phrases are understood to refer to entities s and o, A will bind with $pred([v, s, o])$ to satisfy the parse. The hypothesized definitions are then of the form `define`(v, p, i, j), where i and j are both places of a predicate that relates s to o. Counts and chi-square tests proceed as normal for each such definition found.

Experiment 1: “I” and “You”

For our first experiment, we used the experimental setup of (Gold & Scassellati 2006) using the new Prolog-based

Subject	E. K.	J. H.	M. D.
Facing errors	22.5%	22.5%	30%
Ball location errors	22.5%	17.5%	17.5%
Sound localization errors	2.5%	2.5%	0%
Recognition errors	0%	0%	2.5%
"I" consistent, utterance #	2	17	4
"You" consistent, utterance #	30	36	7

Table 1: Comparison of "I" and "you" learning with different sensory error rates for subject facing, ball location, sound localization, and speech recognition. Most sensory errors were caused by asynchrony between the speech and sensory modules.

TWIG system to learn the words "I" and "you." Two people passed a bright yellow ball back and forth in front of the robot, alternating between the phrases "I got the ball," "You got the ball," and "[name] got the ball" to comment on the action. Subjects were instructed to look at the other person when saying "you" and at the robot when saying the other person's name. (One of the people was always an experimenter, as pairs of subjects left to their own devices tended to speak and act too quickly for the speech recognition system.) All of the words were contained in a small CFG for the purposes of segmentation, but the Prolog system originally only contained the definitions `define(got, has, 1, 2)` and `define(ball, ball, 1)`. The experiment continued for 40 recognized utterances, and was repeated from the beginning with 3 different pairings of people.

For each pair, the words "I," "you," and the names of the two individuals received the correct definitions by the end of the final trial: [[I]] $= tells@1$, [[you]] = $tells@2$, and [[$(name)$]] $= lprop@1$ or $rprop@1$, as appropriate. Figure 4 shows the progress of the definition of "I" for one of these subjects (M. D.), while Table 1 compares the results across subjects, based on error rates. Across subjects, "you" was the most difficult word for the system to learn because it required the correct facing information, correct sound localization, and correct recognition; "I" was much easier to learn because the facing of the subject did not matter. The high number of sensory errors were found to have been caused by timing disparities between the robot's sensory modules and the speech system, but they were not so numerous as to overwhelm the word learning. Errors were classified post hoc based on transcripts, with recognition errors assumed only if another kind of error could not explain the data.

Experiment 2: "Am" and "Are"

For each subject in Experiment 1, the data accumulated in the first experiment was used to initialize the system in the second phase. In this experiment, subjects simply alternated between "I am [name]" and "You are [name]." A ball was again passed back and forth, but this time passing the ball only served to force subjects to pause between utterances. For each subject, the system used only the definitions it learned during the corresponding trial of Experiment 1. The experiment continued until 30 utterances were recognized.

Subject	E. K.	J. H.	M. D.
Facing errors	7%	0%	7%
Sound localization errors	7%	7%	7%
Recognition errors	3%	3%	13%
"Am" consistent, utterance #	1	1	1
"Are" consistent, utterance #	21	2	23

Table 2: Comparison of "am" and "are" learning with different sensory error rates for subject facing, sound localization, and speech recognition.

In all three runs, "am" and "are" were paired with the correct definition of $ident@1,2$. "Am" was apparently easier than "are" because learning it did not require interpreting "you," which involved potentially error-prone facing information. Table 2 compares the results and error rates across subjects. (Facing errors were less common in Experiment 2 because the speakers consistently faced each other.)

Discussion

Learning new words is a useful skill for any robotic system that employs natural language because the robot's environment, and therefore the linguistic demands of the task, may not be known until run-time. However, there are many obvious approaches to the task of word-learning that are incorrect. It is a mistake to assume that all new words will be based on low-level visual functions, because many words refer to function and not form. It is a mistake to assume that the robot will be able to learn new words simply by associating everything in its environment with everything it hears; in psychology, this approach to word learning is called "associationism" and is known to be fallacious (Bloom 2000). It is a mistake to assume that grammar is unimportant, and that proximity to other words is sufficient to define a word; the linking verbs "am" and "are" are generally not in proximity to other words like them, but define a relation between those words. Systems that cannot learn such basic words are likely to fail in more complicated domains.

The ideal word-learning system should be able to leverage all of the information available to the robot at compile time. For applications of AI, as opposed to modeling work, we should care more about the "inductive step" of learning more words, rather than the "base case" of learning first words, because the base case can be preprogrammed. The TWIG system can take advantage of an existing knowledge base, speech recognition system, and semantic information to learn new word groundings, rather than starting from scratch each time.

Though we have presented this work using the language of predicate logic, it should be clear that frame-based semantics (Minsky 1974) fit nicely into this approach as well. Predicates can act as frames, with the predicate arguments serving as slots of the frame. The TWIG system then allows these frame slots to serve as potential new word definitions. For instance, a frame for a car may indicate a place where it is usually found, but not have a word for it;

TWIG could learn through context that this place was called a "garage." TWIG would also function well with planning systems, since planning operators similarly can include slots for objects that aren't necessarily in the robot's vocabulary.

The specific examples in our experiments were chosen to highlight aspects of word learning that TWIG can perform that previous systems could not. For instance, under a "naive associationist" framework, it would be difficult to learn "I" and "you" because every sentence always has a speaker and an addressee; this fact is not true more often when these words are spoken. Rather, it is the method of finding the extension that allowed the system to know when speaking or being addressed was potentially relevant. In the case of "am" and "are," we again showed how a difficult non-visual definition – the identity property – could be learned through the correct use of context. This also demonstrated how the words learned by the system earlier – "I," "you," and the proper names – allowed the system to make sense of sentences that were composed entirely of words it had not understood when the experiments were begun.

The implementation presented here contains a number of limitations, and we shall attempt to evaluate just how limiting these may be in the long term. First, the grammar was simple, and did not contain relative clauses, adjectives, or prepositions. This was primarily to allow us to continue to use Prolog's default search tactic of depth-first search, which can fail to halt when a grammar allows recursive constructions. It also allowed us to sidestep issues of parse ambiguity, which may pose a more serious problem. These issues might be overcome by introducing depth limits to search and weighting evidence by parse likelihoods, respectively.

We also note that our system assumes a segmentation is computed before engaging in parsing and word learning. Ideally, our methods would work equally well with phoneme sequences instead of words, but in practice, accurate phoneme recognition tends to require a language model of word transition probabilities (Jelinek 1997). This fact tends to make abstraction at the phoneme level less appealing, and encourages us to deal with whole words as our semantic targets; luckily, language models generally do not require semantic knowledge.

Two issues that we have addressed since this paper was originally submitted are the ability to deal with predicates that have values associated with them, such as distance and color, and the creation of new definitions that are conjunctions of simpler predicates. The addition of these abilities has allowed TWIG to learn words for simple prepositions such as "above" and "below," and the proximal/distal distinction between "this" and "that." Details of the new method, which organizes word intensions into decision trees, will appear in a future paper.

Despite the recent interest in problems of semantics among roboticists, there has actually been little work that incorporates Montagavian formal semantics with grounded word learning research. We hope that if TWIG demonstrates nothing else, it is that a combination of formal and statistical approaches is necessary to deal with the hard problems of grounded semantics.

Acknowledgments

Support for this work was provided by a National Science Foundation CAREER award (#0238334) and award #0534610 (Quantitative Measures of Social Response in Autism). Some parts of the architecture used in this work was constructed under NSF grants #0205542 (ITR: A Framework for Rapid Development of Reliable Robotics Software) and #0209122 (ITR: Dance, a Programming Language for the Control of Humanoid Robots) and from the DARPA CALO/SRI project. This research was supported in part by a grant of computer software from QNX Software Systems Ltd.

References

Bailey, D. R. 1997. *When Push Comes to Shove: A Computational Model of the Role of Motor Control in the Acquisition of Action Verbs*. Ph.D. Dissertation, Dept. of Computer Science, U.C. Berkeley.

Bloom, P. 2000. *How Children Learn the Meanings of Words*. Cambridge, Massachusetts: MIT Press.

Dowty, D. R.; Wall, R. E.; and Peters, S. 1981. *Introduction to Montague Semantics*. Boston: D. Reidel.

Gold, K., and Scassellati, B. 2006. Grounded pronoun learning and pronoun reversal. In *Proceedings of the 5th International Conference on Development and Learning*.

Jelinek, F. 1997. *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press.

Manning, C. D., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Minsky, M. 1974. A framework for representing knowledge. Technical Report 306, M.I.T. Artificial Intelligence Laboratory.

Montague, R. 1974. *Formal Philosophy*. New Haven, CT: Yale UP.

Pereira, F. C. N., and Shieber, S. M. 1987. *Prolog and Natural-Language Analysis*. Menlo Park, CA: CSLI/SRI International.

Quine, W. V. O. 1960. *Word and Object*. Cambridge, MA: MIT Press.

Regier, T. 1996. *The Human Semantic Potential: Spatial Language and Constrained Connectionism*. Cambridge, MA: MIT Press.

Roy, D. K., and Pentland, A. P. 2002. Learning words from sights and sounds: a computational model. *Cognitive Science* 26:113–146.

Saeed, J. I. 2003. *Semantics*. Malden, MA: Blackwell Publishing, 2nd edition.

Viola, P., and Jones, M. 2004. Robust real-time face detection. *International Journal of Computer Vision* 57(2):137–154.

Yu, C., and Ballard, D. H. 2004. A multimodal learning interface for grounding spoken language in sensory perceptions. *ACM Transactions on Applied Perceptions* 1(1):57–80.